

**AD-A278 227**



**INTERACTIVE  
HIGH RESOLUTION  
FORCE-ON-FORCE  
MODEL**

2

**MAIN BATTLE PROCESSOR  
SOURCE CODE LISTING**

**DTIC**  
**ELECTE**  
**APR 18 1994**  
**S F D**

**SPC 94-282-1A**

**APRIL 1994**

**PREPARED FOR:**

**U.S. ARMY TANK - AUTOMOTIVE COMMAND  
ADVANCED SYSTEMS CONCEPTS DIVISION  
WARREN, MICHIGAN 48397-5000**

**CONTRACT NO. DAAE07-92-C-R044**

This document has been approved  
for public release and sale; its  
distribution is unlimited.

**PREPARED BY:**

**STEVEN M. BUC**

**SYSTEM PLANNING CORPORATION  
1500 WILSON BOULEVARD  
ARLINGTON, VIRGINIA 22209**

**94-11495**

**94 4 15 0 42**

DTIC QUALITY INSPECTED 1

REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE April 1994	3. REPORT TYPE AND DATES COVERED Technical October 1993-March94	
4. TITLE AND SUBTITLE Interactive High Resolution Force-on-Force Model -- Main Battle Processor Source Code Listing			5. FUNDING NUMBERS  C	
6. AUTHOR(S)  Steven M. Buc				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) System Planning Corporation 1500 Wilson Boulevard Arlington, Virginia 22209			8. PERFORMING ORGANIZATION REPORT NUMBER  SPC 94-282-1A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Tank - Automotive Command Advanced Systems Concepts Division Warren, Michigan 48397-5000			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES  Contract No. DAAE07-92-C-R044				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  <p>This document provides the source code listing for the main battle processor for a unique personal computer based interactive force-on-force model. The programming language used is FORTRAN, and this code has been configured to run on 386+ compatible personal computers. The code is currently dimensioned to permit the operator to simulate armored warfare with up to 200 direct fire units on each side. Up to 10 different battlefield systems are permitted within these 200 units, with each system employing up to different on-board weapons.</p> <p>(continued on following page.)</p>				
14. SUBJECT TERMS force-on-force model, interactive, personal computer, artillery, vehicles, direct fire, guided weapons, terrain, obscurants, mine fields, graphics			15. NUMBER OF PAGES 74	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited	

(Abstract continued)

Ten indirect fire support systems may also be modeled to round out the combined arms team. Each artillery system may employ up to 5 different types of ammunition, to include high explosive, submunition dispensing, and guided munitions. The effects of terrain masking are included through the use of a digitized terrain board. Battlefield obscurants, target detection probabilities, and the effects of moving targets on weapon accuracy are also modeled. Mine fields and other obstacles can also be simulated.

The operator directs unit movements and fire orders. The code operates with a variable time step, which can be set as low as one second. The operator can interact with the code operation every sixty cycles. Battlefield activity is displayed as it occurs on the graphics screen. This software was developed in support of studies and analyses of the sequential mission duty cycle/mission profile of a future main battle tank for the year 2015. However, the software is sufficiently flexible to permit the analysis of tradeoffs involving many weapon systems and subsystems on the modern battlefield.

Accession For	
NTIS	CRA&I <input checked="" type="checkbox"/>
DTIC	TAB <input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification .....	
By .....	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



```

UNIT SPEED (TRUE), LOSSES, AMMO REMAINING
C
WRITE ('.') '4-COMMAND PHASE'
GIVE ORDERS AFFECTING:
UNIT MOVEMENT
UNIT FIRES
OBSCURANTS
C
WRITE ('.') '5-BATTLE PHASE'
BY TIME INCREMENT:
MOVE UNITS
SHOOT UNITS
C
WRITE ('.') '6-OUTPUT PHASE'
SELECT OUTPUTS:
UNIT STATUS
LOSSES
CHRONOLOGY OF EVENTS
C
WRITE ('.') '7-CHANGE TIME STEP'
C
WRITE ('.') '0-QUIT'
WRITE ('.')
WRITE ('.') 'ENTER PHASE SELECTION'
WRITE ('.')
READ ('.',ERR=1) IP
RETURN
END
C
SUBROUTINE INPUT
INPUT PHASE
READ:
GRID
CONTOURS
FORCE STRUCTURE
UNIT MOVEMENT
WEAPON SYSTEMS
C
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
& NC(15),NCOL(15),JCON
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& BXYF(200,2),RXYP(200,2),ISPD(200),ISPD(200)
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PXB(180,20),PKR(180,20),
& NRNB(30),NRNR(30),TEMAX(30),TEMAXR(30),TEMNB(30),
& TEMNR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MLB(10,3),MLR(10,3)
COMMON/SHOTS/ENB(200,3),NSHOTB(200,3),NSHOTR(200,3)
COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMMB(200,3),TRMMR(200,3),
& TRMBF(200),TRDESB(200),TRDESR(200)
COMMON/BATTLE/ITIME,ITIME,ITGB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMB(200),
& IAMR(200)
COMMON/ELEVA/CONT(20),IELEV(10000)
COMMON/PUFFS/WSPEED,WDIR,IDUST,EXPUFF,DTPUFF,TENOPUFF,
& IDUST1,IDUST2,NDUST,ENG1,ENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(98,2),TAART(98,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
COMMON/OBSCURE/NGRENB(200,2),NGRENr(200,2),IBESMK(200),
& IBESMK(200)
COMMON/TRACKING/BOREB(10,3),BORER(10,3),IRNGB(30,4),IRNGR(30,4),
& ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)
COMMON/PRIORITY/KPRIB(200,10),KPRIR(200,10)
COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADEr(200)
COMMON/ENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
& ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
COMMON/FLAGS/ITER
COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)
COMMON/COOKERS/NCOOK,KYCOOK(400,2),PBURNB(10),PBURNR(10),
& KBURNB(200),KBURNR(200)
C
CHARACTER*9 BFNAME,RFNAME
CHARACTER*90 LINE
CHARACTER*16 BHIER,RHIER,IBIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
CHARACTER*13 FORM
CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
1002 WRITE ('.')
WRITE ('.') 'TERRAIN ON = 1 OFF = 0'
WRITE ('.')
READ ('.',ERR=1002) ITER
OPEN (1,FILE='GRID.IN')
READ (1,') NX
READ (1,') NY
READ (1,') SCALE
READ (1,') SCFACT
READ (1,') PRFACT
READ (1,') ZFACT
CLOSE (1)
NY=NX
C
OPEN (1,FILE='COLORS.IN')
FORMAT (2I5)
DO 151 I=1,15
151 READ (1,311) NC(I),NCOL(NC(I))
CLOSE (1)
C
INITIALIZE PLOTTING FACTORS
C
XUNIT=11.0/NX*SCFACT
YUNIT=8.5/NY
GXUNIT=11.0/NX*SCFACT
GYUNIT=8.5/NY
XUNIT-GXUNIT/SCALE
YUNIT-GYUNIT/SCALE
211 FORMAT (2F10.4,A5,2F10.4,A5,I5)
IF (ITERLEQ.1) THEN
OPEN (1,FILE='CONTSEGM.DAT')
1=1
101 READ (1,211,ERR=1001) XA,YA,I1,XB,YB,I2,IC(I)
WRITE ('.',211) XA,YA,I1,XB,YB,I2,IC(I)
C
X1(I)=XA*XUNIT
X2(I)=XB*XUNIT
Y1(I)=YA*YUNIT
Y2(I)=YB*YUNIT
1=1+1
JCON=1+1
GO TO 101
C
1001 CONTINUE
CLOSE (1)
C
READ CONTOUR AND GRID ELEVATION INFORMATION
C
OPEN (1,FILE='JDAVIS.DAT')
C.....READ NO. OF DATA POINTS, NO. OF CONTOURS, DATA FORMAT
READ(1,510) NDATA,LC,FORM
C WRITE ('.',510) NDATA,LC,FORM
510 FORMAT(18,I5,A13)
C.....READ THE CONTOUR VALUES
READ(1,FORM) (CONT(I),I=1,LC)
C WRITE ('.',FORM) (CONT(I),I=1,LC)
CLOSE (1)
C
CONVERT CONTOUR INPUT DATA FROM ELEVATIONS IN FEET
TO ELEVATIONS IN KM
DO 514 I=1,LC
514 CONT(I)=CONT(I)*.3048/1000.
ELSE
LC=1
CONT(I)=0.0
ENDIF
C
OPEN (1,FILE='GRID.DAT')
FORMAT (3I8)
DO 513 IX=1,NX
DO 512 IY=1,NY
I=NY*(IX-1)+IY
READ (1,511) NUMX,NUMY,IELEV(I)
IF (ITERLINE.1) IELEV(I)=1
512 CONTINUE
513 CONTINUE
CLOSE (1)
C
READ FORCE STRUCTURE AND MOVEMENT
C
BHIER BLUE FORCE HEIRARCHY
RHIER RED FORCE HEIRARCHY
BUNIT BLUE UNIT NAMES
RUNIT RED UNIT NAMES
IBSYS BLUE SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
IRSYS RED SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
NB NUMBER OF BLUE UNITS
NR NUMBER OF RED UNITS
C
BXYA BLUE UNIT ABSOLUTE X AND Y POSITION
RXYA RED UNIT ABSOLUTE X AND Y POSITION
XG X POSITION IN GRID UNITS
YG Y POSITION IN GRID UNITS
IBSA BLUE UNIT SPEED AND AZIMUTH (ROUNDED TO INTEGER)
IRSA RED UNIT SPEED AND AZIMUTH (ROUNDED TO INTEGER)
C
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
DO 11 I=1,200
BHIER(I)=0
RHIER(I)=0
BUNIT(I)=0
RUNIT(I)=0
IBSYS(I,1)=0
IBSYS(I,2)=0
IRSYS(I,1)=0
IRSYS(I,2)=0
11 CONTINUE
C
OPEN FILES AND READ CURRENT STATUS
C
READ FORCE STRUCTURES
OPEN (1,FILE='BFORCE')
READ (1,3) L1BF
READ (1,3) L2F

```

```

C READ (1,3) L3F
C WRITE (.31) L1RF
C WRITE (.31) L2F
C WRITE (.31) L3F
C FORMAT (A16,A24,2H)
C FORMAT (A90)
C11 FORMAT (1X,A16,A24,2H)
C11 FORMAT (1X,A90)
C11 I=1
C11 READ (1,2,ERR=100) BHIER(I),BUNIT(I),BSYS(1,1),BSYS(1,2)
C11 IF (BSYS(1,1).EQ.0) GO TO 100
C WRITE (.21) BHIER(I),BUNIT(I),BSYS(1,1),BSYS(1,2)
C11 I=I+1
C11 GO TO 10
C11 CONTINUE
C11 NR=I-1
C11 CLOSE (1)
C OPEN (1,FILE='RFORCE')
C READ (1,3) L1RF
C READ (1,3) L2F
C READ (1,3) L3F
C WRITE (.31) L1RF
C WRITE (.31) L2F
C WRITE (.31) L3F
C11 I=1
C11 READ (1,2,ERR=200) RHIER(I),RUNIT(I),RSYS(1,1),RSYS(1,2)
C11 IF (RSYS(1,1).EQ.0) GO TO 200
C WRITE (.21) RHIER(I),RUNIT(I),RSYS(1,1),RSYS(1,2)
C11 I=I+1
C11 GO TO 20
C11 CONTINUE
C11 NR=I-1
C11 CLOSE (1)
C READ MOVEMENT STATUS
C OPEN (1,FILE='BMOVE')
C READ (1,3) L1BM
C READ (1,3) L2M
C READ (1,3) L3M
C WRITE (.31) L1BM
C WRITE (.31) L2M
C WRITE (.31) L3M
C11 FORMAT (A16,A24,4F8.3,2H)
C11 FORMAT (1X,A16,A24,4F8.3,2H)
C11 DO 110 I=1,NR
C11 READ (1,12) BHIER(I),BUNIT(I),BXIA(1,1),BXIA(1,2),BXIO(1,1),
C11 & BXIO(1,2),BSA(1,1),BSA(1,2)
C11 WRITE (.121) BHIER(I),BUNIT(I),BXIA(1,1),BXIA(1,2),BXIO(1,1),
C11 & BXIO(1,2),BSA(1,1),BSA(1,2)
C11 ISPOB(I)=BSA(1,1)
C11 MTYPEB(I)=0
C11 BXYP(1,1)=0
C11 BXYP(1,2)=0
C11 CONTINUE
C11 CLOSE (1)
C OPEN (1,FILE='RMOVE')
C READ (1,3) L1RM
C READ (1,3) L2M
C READ (1,3) L3M
C WRITE (.31) L1RM
C WRITE (.31) L2M
C WRITE (.31) L3M
C11 DO 115 I=1,NR
C11 READ (1,12) RHIER(I),RUNIT(I),RXIA(1,1),RXIA(1,2),RXIO(1,1),
C11 & RXIO(1,2),RSA(1,1),RSA(1,2)
C11 WRITE (.121) RHIER(I),RUNIT(I),RXIA(1,1),RXIA(1,2),RXIO(1,1),
C11 & RXIO(1,2),RSA(1,1),RSA(1,2)
C11 ISPOR(I)=RSA(1,1)
C11 MTYPER(I)=0
C11 RXYP(1,1)=0
C11 RXYP(1,2)=0
C11 CONTINUE
C11 CLOSE (1)
C115
C READ WEAPON SYSTEMS CHARACTERISTICS
C WEAPONS.FOR
C INPUTS WEAPONS CHARACTERISTICS FROM DATA FILES
C FILE STRUCTURE IS DEFINED IN FILE 'WEAPONS.IN'
C11 NBFF # OF BLUE SYSTEM FILES (10 MAX)
C11 NBRF # OF RED SYSTEM FILES (10 MAX)
C11 BFNAME BLUE SYSTEM FILE NAMES (10 MAX)
C11 RFNAME RED SYSTEM FILE NAMES (10 MAX)
C11 NWB # OF BLUE WEAPONS FOR EACH SYSTEM (3 MAX)
C11 NWR # OF RED WEAPONS FOR EACH SYSTEM (3 MAX)
C11 NRWB # OF RANGES FOR EACH BLUE WEAPON PER SYSTEM (6 MAX)
C11 NRWR # OF RANGES FOR EACH RED WEAPON PER SYSTEM (6 MAX)
C11 RB BLUE WEAPON RANGES
C11 RR RED WEAPON RANGES
C11 PKB BLUE PK FOR EACH RED TARGET (EXPOSED/DEFILADE) (10 MAX)
C11 PKR RED PK FOR EACH BLUE TARGET (EXPOSED/DEFILADE) (10 MAX)
C11 NRNOB # SHOTS FOR EACH BLUE WEAPON
C11 NRNOR # SHOTS FOR EACH RED WEAPON
C11 TEMAXB MAX RANGE ENGAGEMENT TIME FOR EACH BLUE WEAPON
C11 TEMAXR MAX RANGE ENGAGEMENT TIME FOR EACH RED WEAPON

```

```

C TEMINB MIN RANGE ENGAGEMENT TIME FOR EACH BLUE WEAPON
C TEMINR MIN RANGE ENGAGEMENT TIME FOR EACH RED WEAPON
C MILB BLUE WEAPON FIRE-AND-FORGET (0) OR MAN-IN-LOOP (1)
C MILR RED WEAPON FIRE-AND-FORGET (0) OR MAN-IN-LOOP (1)
C BORB BLUE BATTLESIGHT RANGES
C BORR RED BATTLESIGHT RANGES
C TOFB BLUE TIME OF FLIGHT TO BATTLESIGHT
C TOFR RED TIME OF FLIGHT TO BATTLESIGHT
C IRNB BLUE WEAPON RANGING/TRACKING THROUGH OBSCURANTS
C IRNR RED WEAPON RANGING/TRACKING THROUGH OBSCURANTS
C ITRB BLUE SYSTEM SEEING THROUGH OBSCURANTS
C ITRR RED SYSTEM SEEING THROUGH OBSCURANTS
C SRCHB SEARCH RATE BLUE SYSTEM
C SRCHR SEARCH RATE RED SYSTEM
C ACELB BLUE SYSTEM ACCELERATION PARAMETERS
C11 1 - MAX SPEED ON FLAT GRADE
C11 2 - TIME FROM ZERO TO MAX SPEED
C11 3 - MAX SPEED ON A GRADE
C11 4 - GRADE FOR THAT MAX SPEED (DEGREES)
C11 5 - TIME FROM ZERO TO THAT MAX SPEED
C ACELR RED SYSTEM ACCELERATION PARAMETERS
C INITIALIZE ARRAYS
C DO 116 JJ=1,180
C RB(JJ)=0.0
C RR(JJ)=0.0
C DO 117 II=1,20
C PKB(JJ,II)=0.0
C PKR(JJ,II)=0.0
C117 CONTINUE
C116 CONTINUE
C DO 118 II=1,30
C NRNOB(II)=0.0
C TEMAXB(II)=0.0
C TEMINB(II)=0.0
C NRNOR(II)=0.0
C TEMAXR(II)=0.0
C TEMINR(II)=0.0
C118 CONTINUE
C DO 119 II=1,10
C DO 120 JJ=1,3
C NWB(II)=0
C NWR(II)=0
C120 CONTINUE
C NRWB(II,JJ)=0
C NWR(II,JJ)=0
C MILB(II,JJ)=1
C MILR(II,JJ)=1
C119 CONTINUE
C301 FORMAT (I2)
C302 FORMAT (A9)
C303 FORMAT (A90)
C501 FORMAT (1X,I2)
C502 FORMAT (1X,A9)
C503 FORMAT (1X,A90)
C OPEN (1,FILE='WEAPONS.IN')
C READ (1,301) NBF
C WRITE (.501) NBF
C DO 310 I=1,NBF
C READ (1,302) BFNAME(I)
C WRITE (.502) BFNAME(I)
C110 CONTINUE
C READ (1,301) NRF
C WRITE (.501) NRF
C DO 312 I=1,NRF
C READ (1,302) RFNAME(I)
C WRITE (.502) RFNAME(I)
C112 CONTINUE
C110 CLOSE (1)
C READ BLUE FILES
C DO 400 I=1,NBF
C OPEN (1,FILE=BFNAME(I))
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C320 FORMAT (I8,3I8)
C520 FORMAT (1X,I8,3I8)
C READ (1,320) NWB(I),(NRWB(I,J),J=1,3)
C WRITE (.520) NWB(I),(NRWB(I,J),J=1,3)
C DO 350 J=1,NWB(I)
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C READ (1,303) LINE
C WRITE (.503) LINE
C321 FORMAT (F8.0,20F8.8)

```

```

821 FORMAT (1X,F7.0,1X,20(F6.3,3X))
DO 340 K=1,NRWB(LJ)
ISR=(I-1)*18+(J-1)*8-K
C WRITE (*,*) ISR,ISR
C WHERE L=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C K=WEAPON RANGE (8 MAX); N=TARGET (2*10 MAX)
READ (1,321) RB(ISR),(PKB(ISR,IST),IST=1,20)
C WRITE (*,321) RB(ISR),(PKB(ISR,IST),IST=1,20)
C WRITE (*,*) RB(22),RB(22)
340 CONTINUE
360 CONTINUE
C WRITE (*,*) I
C WRITE (*,*) RB(ISR),ISR=1,36)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
322 FORMAT (18,10,2F8.2,10,2F8.2)
522 FORMAT (1X,18,10,2X,2(F6.2,3X),1X,18,1X,2F8.2)
DO 380 L=1,NWB(I)
ISROF=(I-1)*3-L
C WHERE L=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C READ (1,322) M,NRND8(ISROF),TEMINB(ISROF),TEMAXB(ISROF),
C & MILB(LL),BORDER(LL),TOFB(LL)
C WRITE (*,322) M,NRND8(ISROF),TEMINB(ISROF),TEMAXB(ISROF),
C & MILB(LL),BORDER(LL),TOFB(LL)
380 CONTINUE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
361 FORMAT (516)
362 FORMAT (1X,516)
DO 361 L=1,NWB(I)
ISROF=(I-1)*3-L
C WHERE L=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C J=CONCEALMENT TYPE (4 MAX)
C READ (1,361) IL,(IRNGB(L),J=1,4)
C WRITE (*,362) IL,(IRNGB(L),J=1,4)
361 CONTINUE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
362 FORMAT (416)
363 FORMAT (1X,416)
C READ (1,362) (ITRKB(LL),L=1,4)
C WRITE (*,363) (ITRKB(LL),L=1,4)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
364 FORMAT (216)
365 FORMAT (1X,216)
C READ (1,364) SRCHB(I),TRATEB(I)
C WRITE (*,365) SRCHB(I),TRATEB(I)
C ACCELERATION PARAMETERS
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,461) (ACELB(LL),L=1,5)
C WRITE (*,462) (ACELB(LL),L=1,5)
C CLOSE (1)
400 CONTINUE
C READ RED FILES
DO 500 L=1,NRF
OPEN (1,FILE=RFNAME(I))
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,320) NWR(I),(NRWR(L),J=1,3)
C WRITE (*,320) NWR(I),(NRWR(L),J=1,3)
DO 460 J=1,NWR(I)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE

```

```

C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
DO 440 K=1,NRWR(LJ)
ISR=(I-1)*18+(J-1)*8-K
C WHERE L=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C K=WEAPON RANGE (8 MAX); N=TARGET (2*10 MAX)
C READ (1,321) RR(ISR),(PKR(ISR,IST),IST=1,20)
C WRITE (*,521) RR(ISR),(PKR(ISR,IST),IST=1,20)
440 CONTINUE
450 CONTINUE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
DO 460 L=1,NWR(I)
ISROF=(I-1)*3-L
C WHERE L=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C READ (1,322) M,NRND8(ISROF),TEMINR(ISROF),TEMAXR(ISROF),
C & MILR(LL),BORER(LL),TOFR(LL)
C WRITE (*,522) M,NRND8(ISROF),TEMINR(ISROF),TEMAXR(ISROF),
C & MILR(LL),BORER(LL),TOFR(LL)
460 CONTINUE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
DO 461 L=1,NWR(I)
ISROF=(I-1)*3-L
C WHERE L=SYSTEM FILE (10 MAX); L=SYSTEM WEAPON (3 MAX);
C J=CONCEALMENT TYPE (4 MAX)
C READ (1,361) IL,(IRNGR(L),J=1,4)
C WRITE (*,362) IL,(IRNGR(L),J=1,4)
461 CONTINUE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,362) (ITRKR(LL),L=1,4)
C WRITE (*,363) (ITRKR(LL),L=1,4)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,364) SRCHR(I),TRATER(I)
C WRITE (*,365) SRCHR(I),TRATER(I)
C ACCELERATION PARAMETERS
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,461) (ACELR(LL),L=1,5)
C WRITE (*,462) (ACELR(LL),L=1,5)
C CLOSE (1)
500 CONTINUE
C READ ELECTROMAGNETIC ENERGY PARAMETERS
OPEN (1,FILE=EMGUN.IN)
C READ (1,3) L1BF
C WRITE (*,31) L1BF
C READ (1,3) L2F
C WRITE (*,31) L2F
531 FORMAT (18)
532 FORMAT (1X,18)
C READ (1,531) NBLUE
C WRITE (*,532) NBLUE
C READ (1,3) L1BF
C WRITE (*,31) L1BF
C READ (1,531) NEMW
C WRITE (*,532) NEMW
C READ (1,3) L1BF
C WRITE (*,31) L1BF
533 FORMAT (216)
504 FORMAT (1X,216)
C READ (1,533) (NEMWN(IEM),IEM=1,NEMW)
C WRITE (*,504) (NEMWN(IEM),IEM=1,NEMW)
C READ (1,3) L1BF
C WRITE (*,31) L1BF
C READ (1,3) L2F
C WRITE (*,31) L2F
505 FORMAT (2F8.3)
506 FORMAT (1X,2F8.3)
DO 507 IEM=1,NEMW
C READ (1,505) (EWP(IEM,JEM),JEM=1,2)
C WRITE (*,506) (EWP(IEM,JEM),JEM=1,2)

```

```

807 CONTINUE
READ (1,3) L1BF
C WRITE (1,3) L1BF
READ (1,3) L1BF
C WRITE (1,3) L1BF
READ (1,3) L1BF
C WRITE (1,3) L1BF
808 FORMAT (5F4.3)
808 FORMAT (1X,5F4.3)
C READ (1,508) (EG(EM),EM=1,5)
C WRITE (1,508) (EG(EM),EM=1,5)
C CLOSE (1)
C
C INITIALIZE BLUE EM WEAPON ENERGIES
AND NUMBER OF SHOTS FOR EACH BLUE AND RED WEAPON
C
IBSYS BLUE SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
IRSYS RED SYSTEM NUMBER AND NUMBER OF SYSTEMS FOR EACH UNIT
NB NUMBER OF BLUE UNITS
NR NUMBER OF RED UNITS
NRNDS # SHOTS FOR EACH BLUE WEAPON
NRNDR # SHOTS FOR EACH RED WEAPON
NWB # OF BLUE WEAPONS FOR EACH SYSTEM (3 MAX)
NWR # OF RED WEAPONS FOR EACH SYSTEM (3 MAX)
* NRNDS(30),NRNDR(30),TEMAB(30),TEMAR(30),TEMNB(30),
ISROF=(1,1)*3-J
C
DO 800 I=1,NB
NSYS=IBSYS(I,1)
IF (NSYS.EQ.NBLUE) THEN
ELSYS(I)=EG(1)
TE(I)=0.0
IF (ELSYS(I).LE.EG(3)) THEN
MODEEM(I)=2
C START CHARGING SYSTEM
ELSE
MODEEM(I)=1
C SYSTEM IS IDLE
ENDIF
ENDIF
NWEAP=NWB(NSYS)
IAMR(I)=0
C SMOKE GRENADES (2 VOLLEYS MAX)
NGRENBI(1,1)=2
NGRENBI(1,2)=0
IBESMK(I)=0
DO 580 J=1,NWEAP
ISROF=(NSYS-1)*3+J
NSHOTBI(I,J)=NRNDS(ISROF)
580 CONTINUE
600 CONTINUE
C
DO 700 I=1,NR
NSYS=IRSYS(I,1)
NWEAP=NWR(NSYS)
IAMR(I)=0
C SMOKE GRENADES (2 VOLLEYS MAX)
NGRENRI(1,1)=2
NGRENRI(1,2)=0
IBESMK(I)=0
DO 680 J=1,NWEAP
ISROF=(NSYS-1)*3+J
NSHOTRI(I,J)=NRNDR(ISROF)
680 CONTINUE
700 CONTINUE
C
INITIALIZE MOVEMENT AND FIRE MODES
C
MANEUVER AND FIRE MODES.
MODEL TRACKS THE CURRENT MODE EACH SYSTEM IS IN,
THE TIME WHEN THE SYSTEM CAN RE-MODE,
AND THE ACCUMULATED TIME IN EACH MODE DURING EACH BATTLE CYCLE
C
THIS MODE INFORMATION IS DUMPED AFTER EACH BATTLE CYCLE
AND TOTAL ACCUMULATED TIME IS DUMPED AT END OF GAME
C
MODEMB CURRENT MANEUVER MODE FOR EACH BLUE SYSTEM (3 MODES)
MODE 1 - DRIVING
MODE 2 - ACCELERATING
MODE 3 - STOPPED
C
MODEMR CURRENT MANEUVER MODE FOR EACH RED SYSTEM
TCMMB CYCLE TIME IN EACH MANEUVER MODE FOR BLUE SYSTEM
TCMMR CYCLE TIME IN EACH MANEUVER MODE FOR RED SYSTEM
TTMMB TOTAL TIME IN EACH MANEUVER MODE FOR BLUE SYSTEM
TTMMR TOTAL TIME IN EACH MANEUVER MODE FOR RED SYSTEM
MODEFB CURRENT FIRE MODE FOR EACH BLUE SYSTEM (9 MODES)
MODE 1 - SEARCHING FOR TARGETS
MODE 2 - ENGAGING A TARGET
MODE 3 - INSUFFICIENT EM ENERGY (HOLD FIRE)
MODE 4 - CREW/GUNNER BALKS AND HOLDS FIRE
MODE 5 - (HOLD FIRE) NO TARGETS WITHIN EFFECTIVE RANGE
MODE 6 - MISSED TARGET RE-ENGAGE
MODE 7 - SURVEILLANCE
MODE 8 - RE-DESIGNATE MAN-IN-LOOP WEAPON
MODE 9 - SACRIFICE SHOT (FIRE-AND-FORGET WEAPON)
C
MODEFR CURRENT FIRE MODE FOR EACH RED SYSTEM
KEWB CURRENT ENGAGEMENT WEAPON ON BLUE SYSTEM
KEWR CURRENT ENGAGEMENT WEAPON ON RED SYSTEM
TCFMB CYCLE TIME IN EACH FIRE MODE FOR BLUE SYSTEM

```

```

C TCFMR CYCLE TIME IN EACH FIRE MODE FOR RED SYSTEM
TTFMB TOTAL TIME IN EACH FIRE MODE FOR BLUE SYSTEM
TTFMR TOTAL TIME IN EACH FIRE MODE FOR RED SYSTEM
TRMBB TIME AT WHICH BLUE SYSTEM CAN RE-MODE MANEUVER
TRMBR TIME AT WHICH BLUE SYSTEM CAN RE-MODE FIRE
TRMMR TIME AT WHICH RED SYSTEM CAN RE-MODE MANEUVER
TRMFR TIME AT WHICH RED SYSTEM CAN RE-MODE FIRE
C
COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
* TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
* MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
* TTFMB(200,9),TTFMR(200,9),TRMBB(200),TRMBR(200),TRMMR(200),
* TRMFR(200),TRDESB(200),TRDESR(200)
C
DO 800 I=1,NB
MODEMB(I)=1
IF (IBSA(I,1).EQ.0) MODEMB(I)=3
TTMMB(I,1)=0.0
TTMMB(I,2)=0.0
TTMMB(I,3)=0.0
MODEFB(I)=7
AZTB(I)=IBSA(I,2)
TTFMB(I,1)=0.0
TTFMB(I,2)=0.0
TTFMB(I,3)=0.0
TTFMB(I,4)=0.0
TTFMB(I,5)=0.0
TTFMB(I,6)=0.0
TTFMB(I,7)=0.0
TRMMB(I)=0.0
TRMFB(I)=0.0
C TOTAL TIMES FOR EACH CYCLE ARE INITIALIZED GOING INTO
BATTLE PHASE
C
INITIALIZE TERRAIN GRADE FOR EACH VEHICLE FOR ACCELERATION
DETERMINATION
KSIDE=1
CALL GRADE (KSIDE,I)
C
INITIALIZE ACTUAL VEHICLE SPEED
ISPOB(I)=IBSA(I,1)
C
800 CONTINUE
C
DO 850 I=1,NR
MODEMR(I)=1
IF (IRSA(I,1).EQ.0) MODEMR(I)=3
TTMMR(I,1)=0.0
TTMMR(I,2)=0.0
TTMMR(I,3)=0.0
MODEFR(I)=7
AZTR(I)=IRSA(I,2)
TTFMR(I,1)=0.0
TTFMR(I,2)=0.0
TTFMR(I,3)=0.0
TTFMR(I,4)=0.0
TTFMR(I,5)=0.0
TTFMR(I,6)=0.0
TTFMR(I,7)=0.0
TRMMR(I)=0.0
TRMFR(I)=0.0
C INITIALIZE TERRAIN GRADE FOR EACH VEHICLE FOR ACCELERATION
DETERMINATION
KSIDE=1
CALL GRADE (KSIDE,I)
C
INITIALIZE ACTUAL VEHICLE SPEED
ISPDR(I)=IRSA(I,1)
C
850 CONTINUE
C
WRITE (1,1) '1'
WRITE (1,1) (RB(ISR),ISR=1,36)
WRITE (1,1) (RR(ISR),ISR=1,36)
C
WIND CONDITIONS
855 WRITE (1,1)
WRITE (1,1) 'WIND CONDITIONS'
WRITE (1,1) '1 - RANDOM WIND - 15 KNOTS MAX, ANY DIRECTION'
WRITE (1,1) '2 - ENTER WIND CONDITIONS'
WRITE (1,1)
READ (1,1) ERR-855) IWIND
IF (IWIND.LT.1.OR.IWIND.GT.2) GO TO 865
IF (IWIND.EQ.1) THEN
WRAND=RND0
IWSPEED=NINT(15.0*WRAND)
DRAND=RND0
IWDIR=NINT(DRAND*360.0)
WRITE (1,1)
WRITE (1,1) 'RANDOM WIND'
WRITE (1,1) 'IWSPEED: KNOTS going towards AZIMUTH: IWDIR'
WRITE (1,1)
ELSE
WRITE (1,1)
WRITE (1,1) 'ENTER WIND SPEED (KNOTS max 15) and AZIMUTH'
WRITE (1,1)
READ (1,1) ERR-855) WRAND,DRAND
IF (WRAND.LT.15.) WRAND=15.
IF (DRAND.LT.0.0) DRAND=DRAND+360.
IF (DRAND.GT.360.0) DRAND=DRAND-360.0

```



```

      IWSPEED=INT(WRAND)
      IWDIR=INT(DRAND)
      WRITE (1,1)
      WRITE (1,1) IWSPEED, ' KNOTS going towards AZIMUTH ',IWDIR
      WRITE (1,1)
      ENDIF
C
C DUST CONDITIONS
C
885 WRITE (1,1)
WRITE (1,1) DUST CONDITIONS
WRITE (1,1) 1 - YES - PLAY DUST
WRITE (1,1) 2 - NO - DO NOT PLAY DUST
WRITE (1,1)
READ (1,1) ERR=885) IDUST
IF (IDUST.LT.1.OR.IDUST.GT.2) GO TO 885

C INITIALIZE SMOKE AND DUST ARRAYS
C
C DUST AND ENGINE SMOKE CLOUDS ARE ACCOUNTED FOR AS DISCRETE
C PUFFS. GRENADE AND ARTILLERY SMOKE ARE MODELED AS CURTAINS.
C PUFFS ARE HEMISPHERICAL FOR PURPOSES OF EXPANSION AND WIND DRIFT.
C CURTAINS DO NOT EXPAND, BUT DO DRIFT ON THE WIND.
C PUFFS CAN BE GENERATED IN 10 SECOND INCREMENTS. PUFFS AND
C CURTAINS ARE MOVED ON THE WIND ONLY EVERY 10 SECONDS.
C PUFFS AND CURTAINS DISSIPATE EVERY 60 SECONDS. PUFFS EXPAND
C AT A RATE OF 5 FEET PER SECOND.
EXPUFF=5.0*3048/1000.
DTPUFF=10.0
TENDPUFF=60.0

C DUST ARRAYS
C ACTIVE CLOUDS: BEGINNING NUMBER = 1; ENDING NUMBER = 1
PTDUST (2400,3) - POSITION X,Y AND TIME FOR DUST PUFFS
IDUST1=0
IDUST2=0
NDUST=0
NDUSTMX=2400

C ENGINE SMOKE ARRAYS
C ACTIVE CLOUDS: BEGINNING NUMBER = 1; ENDING NUMBER = 1
PTENG (2400,3) - POSITION X,Y AND TIME FOR ENGINE SMOKE PUFFS
IENG1=0
IENG2=0
NENG=0
NENGMX=2400

C SMOKE GRENADE CURTAINS
C ACTIVE CLOUDS: BEGINNING NUMBER = 1; ENDING NUMBER = 1
PSMK (400,2) - POSITION X,Y FOR SMOKE GRENADE CURTAIN
TASMK (400,2) - TIME AND AZIMUTH FOR SMOKE GRENADE CURTAIN
ISMK1=0
ISMK2=0
NSMK=0
NSMKMX=400

C ARTILLERY SMOKE CURTAINS
C ACTIVE CLOUDS: BEGINNING NUMBER = 1; ENDING NUMBER = 1
PART (99,2) - POSITION X,Y FOR ARTILLERY SMOKE CURTAIN
TART (99,2) - TIME AND RADIUS FOR ARTILLERY SMOKE CURTAIN
IART1=0
IART2=0
NART=0
NARTMX=99

C INITIALIZE TARGET PRIORITIZATION
C
DO 901 K=1,200
DO 902 KJ=1,10
KPRIB(K,KJ)=1
KPRIR(K,KJ)=1
902 CONTINUE
901 CONTINUE

C READ TARGET DETECTION SIGNATURE DATA
CALL RSGNAT
C READ ACTIVE PROTECTION SYSTEM DATA
CALL READAPS
C READ PROBABILITY OF VEHICLE BURNING DATA AND INITIALIZE ARRAYS
CALL READCOOK
C READ MINEFIELD DATA
CALL READMINES
C READ ARTILLERY SYSTEM DATA
CALL READARTY
C INITIALIZE ARTILLERY AVAILABILITY
CALL ARTINIT

RETURN
END

SUBROUTINE MAPIT
C
C MAP FOR
C PLOTS GRID
C THEN OVERLAYS CONTOURS
C
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),JC(1100),
& NC(15),NCO(15),JCON
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),

```

```

& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXVA(200,2),BSA(200,2),RSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/MAINEUVER/BXYO(200,2),RXVO(200,2),MTYPEB(200),MTYPEP(200),
& BXP(200,2),RXP(200,2),ISPB(200),ISPP(200)
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRNB(30),NRNR(30),TEMAB(30),TEMAR(30),TEMINB(30),
& TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MLB(10,3),MLR(10,3)
COMMON/FLAGS/ITER

C CHARACTER'S BFNAME,RFNAME
C CHARACTER'S LINE
C CHARACTER'S BHIER,RHIER,IHIER
C CHARACTER'S BUNIT,RUNIT
C CHARACTER'S L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C CHARACTER'S C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C CHARACTER'S WELL

C
120 CALL ZOOMIN (WELL)
IF (WELL.EQ.0) GO TO 200
121 WRITE (1,1)
WRITE (1,1) 'ENTER MAPPING OPTIONS'
WRITE (1,1) 'GRID,CONTOURS,UNITS,OBSCURANTS (1=YES,0=NO)'
WRITE (1,1)
READ (1,1) ERR=121) IGRID,ICONF,IUNIT,IOBSC
CALL PLOTS (0,1,0)
IF (ICONF.EQ.1.AND.ITER.EQ.1) CALL CONTPLOT (JCON)
IF (IGRID.EQ.1) CALL GRID2
IF (IUNIT.EQ.1) CALL PLOTMINES
IF (IUNIT.EQ.1) CALL UNITS
IF (IOBSC.EQ.1) THEN
CALL PDUST
C PLOT ENGINE SMOKE CLOUDS
CALL PENG
C PLOT SMOKE GRENADE CURTAINS
CALL PSMKCURTAINS
C PLOT ARTILLERY SMOKE CURTAINS
CALL PARTSMK
ENDIF
READ (1,1)
CALL PLOT (0,0,999)
GO TO 120
200 RETURN
END

SUBROUTINE GRID2
GRID.FOR
C
C GRIDS MAP WITH SQUARES
C OPENS GRID.DAT FILE FOR ELEVATION DATA AT EACH GRID CENTER
C
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT

NX NUMBER OF X GRIDS
NY NUMBER OF Y GRIDS (SHOULD EQUAL NX)
SCALE REAL WORLD SCALE FACTOR FOR EACH GRID
SCFACT SCREEN SCALING FACTOR FOR X DIRECTION
PRFACT PRINTER SCALING FACTOR FOR X DIRECTION
ZFACT ZOOM FACTOR FOR SIZE OF GRID ON SCREEN

ISKIP=0

DRAW A SQUARE FOR EACH GRID LOCATION
C
CALL PLOTS (0,1,0)
XUNIT=11.0/NX*SCFACT
YUNIT=8.5/NY
CALL FACTOR (ZFACT)
IF (XLEN.GT.0) CALL FACTOR (ZFACTN)
DO 100 IX=1,NX
X=X-IX*XUNIT-XUNIT/2.
DO 50 IY=1,NY
Y=Y-IY*YUNIT-YUNIT/2.
C MOVE TO GRID SQUARE CENTER BOTTOM LEFT CORNER
IF (XLEN.GT.0) CALL ZOOMIT (X,Y,ISKIP)
IF (ISKIP.EQ.1) GO TO 50
CALL PLOT (X,Y,3)
C DRAW FOUR SIDES OF SQUARE
CALL PLOT (0,YUNIT,12)
CALL PLOT (XUNIT,0,12)
CALL PLOT (0,-YUNIT,12)
CALL PLOT (-XUNIT,0,12)
50 CONTINUE
100 CONTINUE
C READ (1,1)
RETURN
END

SUBROUTINE CONTPLOT (K)
CONTPLOT.FOR
C
C PLOTS CONTOUR MAP AS GENERATED BY SOFTWARE
C OPENS CONTSSEG.DAT FILE FOR CONTOUR SEGMENTS
C

```

7

```

C ENTER MOVEMENT AND FIRE CONTROL ORDERS FOR UNITS
C
COMMON/GRID/KNY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/TOURS/1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
C & NC(15),NCOL(15),ICOL
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/FORCE/BIER(200),RBIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,L2RF,L3RF,NS,NR
COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NY,J(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRND(30),NRNR(30),TEMAXB(30),TEMAXR(30),TE(30),
& TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MILB(10,3),MILR(10,3)
COMMON/ANE/VER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200),
& BXYP(200,2),RXP(200,2),ISPOB(200),ISPOR(200)
COMMON/MODE/3/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
& MODEFR(200),KEYWB(200),KEYWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFR(200),TRMMR(200),
& TRMFR(200),TRDES(200),TRDESR(200)
COMMON/DEFLADE/IDFB(200),IDFR(200)
COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)
COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)
COMMON/AAVAL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
C
C CHARACTER'S BFNAME,RFNAME
C CHARACTER'S LINE
C CHARACTER'S BIER,RBIER,IER
C CHARACTER'S BUNIT,RUNIT
C CHARACTER'S L1BF,L2F,L3F,L1RF,L2M,L3M,L1RM
C CHARACTER'S C2
C CHARACTER'S C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
C MOVEMENT ORDERS
C
PL=3,14159
KSIDE=0
10 WRITE (C,*)
WRITE (C,*) 'enter BLUE or RED, or QUIT'
WRITE (C,*)
READ (C,*,ERR=10) C2
IF (C2(1:1).EQ.'B') KSIDE=1
IF (C2(1:1).EQ.'R') KSIDE=2
IF (C2(1:1).EQ.'Q') GO TO 1000
IF (KSIDE.EQ.0) GO TO 10
C
5 WRITE (C,*)
WRITE (C,*) 'CURRENT STATUS'
FORMAT (1X,A30)
11 IF (C2(1:1).EQ.'B') WRITE (C,11) L1BM
IF (C2(1:1).EQ.'R') WRITE (C,11) L1RM
WRITE (C,*) UNIT HIERARCHY IDENTITY POSITION
& OBJECTIVE SPD AZIN
IF (C2(1:1).EQ.'B') THEN
C BLUE
KK=0
DO 130 K=1,NB
121 FORMAT (1X,13,1X,A16,A24,4(F8,3,1X),13,1X,13)
IF (IRSYS(K,2).GE.1) THEN
WRITE (C,121) K,BIER(K),BUNIT(K),BXYA(K,1),BXYA(K,2),BXYO(K,1),
& BXYO(K,2),IBSA(K,1),IBSA(K,2)
KK=KK+1
ENDIF
IF (KK.EQ.20) THEN
READ (C,*)
KK=0
ENDIF
130 CONTINUE
C
ELSE
RED
KK=0
DO 140 K=1,NR
IF (IRSYS(K,2).GE.1) THEN
WRITE (C,121) K,RBIER(K),RUNIT(K),RXYA(K,1),RXYA(K,2),RXYO(K,1),
& RXYO(K,2),IRSA(K,1),IRSA(K,2)
KK=KK+1
ENDIF
IF (KK.EQ.20) THEN
READ (C,*)
KK=0
ENDIF
140 CONTINUE
ENDIF
141 WRITE (C,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
WRITE (C,*)
READ (C,*,ERR=141) ICOMMAND
IF (ICOMMAND.EQ.0) GO TO 5
IF (ICOMMAND.EQ.-1) GO TO 10
142 WRITE (C,*) 'ENTER NUMBER RANGE OF SUBORDINATES TO FOLLOW'
WRITE (C,*) 'THIS ORDER (#START #END) (0 0 FOR NONE)'
WRITE (C,*)
READ (C,*,ERR=142) NSUB1,NSUB2
IF (NSUB1.LE.0) THEN
NSUB1=0
NSUB2=0
ENDIF
IF (NSUB2.LE.0) THEN

```

```

NSUB1=0
NSUB2=0
ENDIF
C
150 WRITE (C,*)
WRITE (C,*) 'MOVEMENT TYPE'
WRITE (C,*) '1-FOLLOW ME'
WRITE (C,*) '2-COME ON LINE'
WRITE (C,*) '3-TURN'
WRITE (C,*) '4-SPEED CHANGE'
WRITE (C,*) '5-DEFILADE'
WRITE (C,*) '6-BRECH OBSTACLES'
WRITE (C,*) '7-QUIT'
READ (C,*,ERR=150) MTYPE
WRITE (C,*)
IF (MTYPE.LE.0.OR.MTYPE.GE.8) GO TO 150
IF (MTYPE.EQ.7) GO TO 5
IF (MTYPE.EQ.5) CALL DIGIN
IF (MTYPE.EQ.8) CALL BREECH
C
IF (MTYPE.EQ.1) THEN
C FOLLOW ME - ALL UNITS TURN AT THEIR RELATIVE PIVOT POINTS
C AND THEN HEAD TO NEW X,Y OBJECTIVE, WHERE THEY WILL STOP
151 WRITE (C,*) 'ENTER THE RANGE TO THE PIVOT POINT ON THE CURRENT'
WRITE (C,*) 'LINE OF ADVANCE AND THE RANGE AND AZIMUTH TO THE'
WRITE (C,*) 'NEW OBJECTIVE POINT FOR THE COMMAND UNIT'
WRITE (C,*)
READ (C,*,ERR=151) RANGE,RANGE0,AZO
IF (KSIDE.EQ.1) THEN
MTYPEB(ICOMMAND)=1
IDFB(ICOMMAND)=0
ELSE
MTYPER(ICOMMAND)=1
IDFR(ICOMMAND)=0
ENDIF
IF (KSIDE.EQ.1) THEN
IAZ=IBSA(ICOMMAND,2)
X0=BXYA(ICOMMAND,1)
Y0=BXYA(ICOMMAND,2)
ELSE
IAZ=IRSA(ICOMMAND,2)
X0=RXYA(ICOMMAND,1)
Y0=RXYA(ICOMMAND,2)
ENDIF
AZIN=IAZ
CALL FDIR (AZIN,AZIN)
C
C FIND COMMAND UNIT PIVOT POINT
XP=X0+RANGE*COS(AZIN*PI/180.)
YP=Y0+RANGE*SIN(AZIN*PI/180.)
C
C FIND COMMAND UNIT OBJECTIVE POINT
CALL FDIR (AZO,AZOUT)
XO=XP+RANGE0*COS(AZOUT*PI/180.)
YO=YP+RANGE0*SIN(AZOUT*PI/180.)
C
C STORE THIS INFORMATION SO WHEN PIVOT POINT IS REACHED
C UNIT WILL TURN AND HEAD TO NEW OBJECTIVE
IF (KSIDE.EQ.1) THEN
BXYP(ICOMMAND,1)=XP
BXYP(ICOMMAND,2)=YP
BXYO(ICOMMAND,1)=XO
BXYO(ICOMMAND,2)=YO
ELSE
RXP(ICOMMAND,1)=XP
RXP(ICOMMAND,2)=YP
RXYO(ICOMMAND,1)=XO
RXYO(ICOMMAND,2)=YO
ENDIF
C
C FIND RELATIVE PIVOT POINT AZIMUTH FOR DETERMINING SUBORDINATE
C UNITS' RANGE TO THEIR PIVOT POINT
C BISECT ANGLE BETWEEN OLD AZIMUTH AND NEW AZIMUTH AT PIVOT POINT
AZINCL=-(AZIN-AZOUT)/2
AZBISECT=AZOUT+AZINCL
SLOPE=TAN(AZBISECT/180.*PI)
B=YP-SLOPE*XP
C
C FOR EACH SUBORDINATE, PIVOT POINT IS ON THIS LINE
C FIND INTERSECTION OF CURRENT AZIMUTH AND LOCATION WITH
C THIS PIVOT AZIMUTH AND PIVOT POINT
DO 170 I=NSUB1,NSUB2
NSKIP=0
IF (KSIDE.EQ.1) THEN
IDFB(I)=0
MTYPEB(I)=1
X1=BXYA(I,1)
Y1=BXYA(I,2)
IAZ=IBSA(I,2)
IF (IRSYS(I,2).LT.1) NSKIP=1
ELSE
IDFR(I)=0
MTYPER(I)=1

```

```

X1=RXVA(1)
Y1=RYVA(2)
IAZ=IRSA(2)
IF (IRSYS(2),LT.1) NSKIP=1
ENDIF
C IF (NSKIP.EQ.1) GO TO 170
C FIND EACH UNIT'S ORIGINAL LONGITUDINAL DISTANCE FROM THE
C COMMAND UNIT ADD OR SUBTRACT THIS DISTANCE FROM RANGE TO
C NEW OBJECTIVE
DELX=X1-X0
DELY=Y1-Y0
IF (AZIN.GE.360) AZIN=AZIN-360.
IF (AZIN.LT.0) AZIN=AZIN+360.
C FIND DISTANCE BETWEEN UNIT LOCATION AND AZIMUTH LINE THROUGH
C COMMAND UNIT
SLOPEL=TAN(AZIN/180.*PI)
BL=Y0-SLOPEL*X0
SLOPEP=SLOPEL
BP=Y1-SLOPEP*X1
C FIND INTERSECTION POINT
XINT=(BP-BL)/(SLOPEL-SLOPEP)
YINT=BL+SLOPEL*XINT
C FIND DISTANCES
DISTL2=(XINT-X1)**2+(YINT-Y1)**2
DIST02=(DELX)**2+(DELY)**2
IF (DIST02.LT.DISTL2) THEN
DIST=0.0
ELSE
DIST=(DIST02-DISTL2)**.5
ENDIF
C IF (AZIN.EQ.90.) DIST0=DELY
IF (AZIN.EQ.270.) DIST0=-DELY
IF (AZIN.EQ.0.) DIST0=DELX
IF (AZIN.EQ.180.) DIST0=-DELX
IF (AZIN.GT.0.0.AND.AZIN.LT.90.0) THEN
IF (DELX.GT.0.) THEN
DIST0=DIST
ELSE
DIST0=-DIST
ENDIF
ENDIF
IF (AZIN.GT.90.0.AND.AZIN.LT.270.0) THEN
IF (DELX.GT.0.) THEN
DIST0=-DIST
ELSE
DIST0=DIST
ENDIF
ENDIF
IF (AZIN.GT.270.0.AND.AZIN.LT.360.0) THEN
IF (DELX.GT.0.) THEN
DIST0=DIST
ELSE
DIST0=-DIST
ENDIF
ENDIF
C RANGE=RANGE0+DIST0
C AZIN1=IAZ
CALL FDIR (AZIN1,AZOUT1)
SLOPE1=TAN(AZOUT1/180.*PI)
B1=Y1-SLOPE1*X1
C INTERSECTION IS PIVOT POINT
XINT=(B1-BL)/(SLOPE1-SLOPEL)
YINT=BL+SLOPEL*XINT
IF (KSIDE.EQ.1) THEN
BXYP(1)=XINT
BXYP(2)=YINT
XQ=XINT+RANGE*COSE(AZOUT/180.*PI)
YQ=YINT+RANGE*SIN(AZOUT/180.*PI)
BXVO(1)=XQ
BXVO(2)=YQ
ELSE
RXYP(1)=XINT
RXYP(2)=YINT
XQ=XINT+RANGE*COSE(AZOUT/180.*PI)
YQ=YINT+RANGE*SIN(AZOUT/180.*PI)
RXVO(1)=XQ
RXVO(2)=YQ
ENDIF
170 CONTINUE
ENDIF
C IF (MTYPE.EQ.2) THEN
C COME ON LINE - ALL UNITS FORM UP AND STOP ON A LINE DESCRIBED
C BY THE PIVOT POINT AND LINE AZIMUTH
171 WRITE (",") 'ENTER THE RANGE TO THE PIVOT POINT ON THE CURRENT'
WRITE (",") 'LINE OF ADVANCE AND AN AZIMUTH'
WRITE (",") 'DESCRIBING THE LINE TO FORM UP ON'
WRITE (",")
READ (",ERR=171") RANGE0,AZIMUTH
CALL FDIR (AZIMUTH,AZOUT)
C FIND THE DISTANCE EACH UNIT IS FROM THE PIVOT POINT
C FIND NEW POSITION ON THE LINE AT THE SAME DISTANCE
C TAKE STRAIGHT PATH TO THIS POINT
IF (KSIDE.EQ.1) THEN
DEFB(ICOMMAND)=0
X0=BXVA(ICOMMAND,1)
Y0=BYVA(ICOMMAND,2)
IAZ=IRSA(ICOMMAND,2)
AZIN0=IAZ
CALL FDIR (AZIN0,AZIN)
C FIND PIVOT POINT
XP=X0+RANGE0*COSE(AZIN/180.*PI)
YP=Y0+RANGE0*SIN(AZIN/180.*PI)
X2=XP+RANGE0*COSE(AZIMUTH/180.*PI)
Y2=YP+RANGE0*SIN(AZIMUTH/180.*PI)
BXVO(ICOMMAND,1)=X2
BXVO(ICOMMAND,2)=Y2
ELSE
DEFB(ICOMMAND)=0
X0=RXVA(ICOMMAND,1)
Y0=RYVA(ICOMMAND,2)
IAZ=IRSA(ICOMMAND,2)
AZIN0=IAZ
CALL FDIR (AZIN0,AZIN)
C FIND PIVOT POINT
XP=X0+RANGE0*COSE(AZIN/180.*PI)
YP=Y0+RANGE0*SIN(AZIN/180.*PI)
X2=XP+RANGE0*COSE(AZIMUTH/180.*PI)
Y2=YP+RANGE0*SIN(AZIMUTH/180.*PI)
RXVO(ICOMMAND,1)=X2
RXVO(ICOMMAND,2)=Y2
ENDIF
TOP=Y2-Y0
BOT=X2-X0
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
IF (BOT.LT.0.0) THEN
IF (TOP.LT.0.0) DIR=DIR+180.
IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
IF (TOP.LT.0.0) DIR=360.-DIR
ENDIF
ENDIF
CALL FAZ (DIR,AZIN)
IF (KSIDE.EQ.1) THEN
IRSA(ICOMMAND,2)=INT(AZIN)
MTYPEB(ICOMMAND)=2
ELSE
IRSA(ICOMMAND,2)=INT(AZIN)
MTYPEP(ICOMMAND)=2
ENDIF
DO 180 L=NSUB1,NSUB2
NSKIP=0
IF (KSIDE.EQ.1) THEN
DEFB(L)=0
IF (IRSYS(2),LT.1) NSKIP=1
X1=BXVA(L,1)
Y1=BYVA(L,2)
ELSE
DEFB(L)=0
IF (IRSYS(2),LT.1) NSKIP=1
X1=RXVA(L,1)
Y1=RYVA(L,2)
ENDIF
C IF (NSKIP.EQ.1) GO TO 180
C FIND EACH UNIT'S ORIGINAL LONGITUDINAL DISTANCE FROM THE
C COMMAND UNIT ADD OR SUBTRACT THIS DISTANCE FROM RANGE TO
C PIVOT POINT
DELX=X1-X0
DELY=Y1-Y0
IF (AZIN.GE.360) AZIN=AZIN-360.
IF (AZIN.LT.0) AZIN=AZIN+360.
C FIND DISTANCE BETWEEN UNIT LOCATION AND AZIMUTH LINE THROUGH
C COMMAND UNIT
SLOPEL=TAN(AZIN/180.*PI)
BL=Y0-SLOPEL*X0
SLOPEP=SLOPEL
BP=Y1-SLOPEP*X1
C FIND INTERSECTION POINT
XINT=(BP-BL)/(SLOPEL-SLOPEP)
YINT=BL+SLOPEL*XINT
C FIND DISTANCES
DISTL2=(XINT-X1)**2+(YINT-Y1)**2
DIST02=(DELX)**2+(DELY)**2
IF (DIST02.LT.DISTL2) THEN
DIST=0.0
ELSE
DIST=(DIST02-DISTL2)**.5
ENDIF
C IF (AZIN.EQ.90.) DIST0=DELY
IF (AZIN.EQ.270.) DIST0=-DELY
IF (AZIN.EQ.0.) DIST0=DELX
IF (AZIN.EQ.180.) DIST0=-DELX
IF (AZIN.GT.0.0.AND.AZIN.LT.90.0) THEN
IF (DELX.GT.0.) THEN
DIST0=DIST
ELSE
DIST0=-DIST
ENDIF
ENDIF
IF (AZIN.GT.90.0.AND.AZIN.LT.270.0) THEN
IF (DELX.GT.0.) THEN
DIST0=-DIST
ELSE
DIST0=DIST
ENDIF
ENDIF
IF (AZIN.GT.270.0.AND.AZIN.LT.360.0) THEN
IF (DELX.GT.0.) THEN
DIST0=DIST
ELSE
DIST0=-DIST
ENDIF
ENDIF
IF (AZIN.GT.270.0.AND.AZIN.LT.360.0) THEN
IF (DELX.GT.0.) THEN
DIST0=DIST
ELSE
DIST0=-DIST
ENDIF
ENDIF

```

```

      ELSE DISTO=DIST
    ENDF
  ENDF
C
  RANGE=RANGE-DISTO
  IF (KSIDE.EQ.1) THEN
    X2=XP+RANGE*COS(AZIMUTH/180.*PI)
    Y2=YP+RANGE*SIN(AZIMUTH/180.*PI)
    BXYO(1,1)=X2
    BXYO(1,2)=Y2
  ELSE
    X2=XP+RANGE*COS(AZIMUTH/180.*PI)
    Y2=YP+RANGE*SIN(AZIMUTH/180.*PI)
    RXYO(1,1)=X2
    RXYO(1,2)=Y2
  ENDF
C
  TOP=Y2-Y1
  BOT=X2-X1
  IF (BOT.EQ.0.0) BOT=BOT+.001
  DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
  IF (BOT.LT.0.0) THEN
    IF (TOP.LT.0.0) DIR=DIR+180.
    IF (TOP.GT.0.0) DIR=180.-DIR
  ELSE
    IF (TOP.LT.0.0) DIR=360.-DIR
  ENDF
  CALL FAZ (DIR,AZIN)
  IF (KSIDE.EQ.1) THEN
    IBSA(1,2)=INT(AZIN)
    MTYPEB(1)=2
  ELSE
    IRSA(1,2)=INT(AZIN)
    MTYPEB(1)=2
  ENDF
180 CONTINUE
ENDF
IF (MTYPE.EQ.3) THEN
  TURN - ALL UNITS TURN TO THE NEW AZIMUTH AND PROCEED TO NEW
  OBJECTIVE POINT DESCRIBED BY THE DISTANCE ALONG THAT NEW AZIMUTH
C
  WRITE ('.') 'ENTER NEW AZIMUTH AND RANGE TO OBJECTIVE FOR'
  WRITE ('.') 'COMMAND UNIT'
  READ ('.',ERR=181) AZIMUTH,RANGE
  CALL FDIR (AZIMUTH,AZOUT)
  IF (KSIDE.EQ.1) THEN
    IDEFB(ICOMMAND)=0
    X1=BXYA(ICOMMAND,1)
    Y1=BXYA(ICOMMAND,2)
    X2=X1+RANGE*COS(AZOUT/180.*PI)
    Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
    BXYO(ICOMMAND,1)=X2
    BXYO(ICOMMAND,2)=Y2
    IBSA(ICOMMAND,2)=INT(AZIMUTH)
    MTYPEB(ICOMMAND)=3
  ELSE
    IDEFR(ICOMMAND)=0
    X1=RXYA(ICOMMAND,1)
    Y1=RXYA(ICOMMAND,2)
    X2=X1+RANGE*COS(AZOUT/180.*PI)
    Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
    RXYO(ICOMMAND,1)=X2
    RXYO(ICOMMAND,2)=Y2
    IRSA(ICOMMAND,2)=INT(AZIMUTH)
    MTYPEB(ICOMMAND)=3
  ENDF
  DO 190 L=NSUB1,NSUB2
    IF (KSIDE.EQ.1) THEN
      IDEFB(I)=0
      IF (IBSYS(1,2).GE.1) THEN
        X1=BXYA(I,1)
        Y1=BXYA(I,2)
        X2=X1+RANGE*COS(AZOUT/180.*PI)
        Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
        BXYO(I,1)=X2
        BXYO(I,2)=Y2
        IBSA(I,2)=INT(AZIMUTH)
        MTYPEB(I)=3
      ENDF
    ELSE
      IDEFR(I)=0
      IF (IRSYS(1,2).GE.1) THEN
        X1=RXYA(I,1)
        Y1=RXYA(I,2)
        X2=X1+RANGE*COS(AZOUT/180.*PI)
        Y2=Y1+RANGE*SIN(AZOUT/180.*PI)
        RXYO(I,1)=X2
        RXYO(I,2)=Y2
        IRSA(I,2)=INT(AZIMUTH)
        MTYPEB(I)=3
      ENDF
    ENDF
  CONTINUE
  ENDF
C
  IF (MTYPE.EQ.4) THEN
    SPEED CHANGE
  C
  181 WRITE ('.') 'ENTER DESIRED SPEED'
  WRITE ('.')
  READ ('.',ERR=181) SPEED

```

```

  IF (SPEED.LT.0.0) GO TO 191
  IF (KSIDE.EQ.1) THEN
    C
    C TO ACCOUNT FOR ACCELERATION PARAMETERS OF VEHICLE
    C THIS DESIRED SPEED IS CONTINUOUSLY MATCHED AGAINST THE
    C MOBILITY PARAMETERS ENTERED FOR THIS SYSTEM AND AN ACTUAL
    C SPEED IS DETERMINED FOR EACH TIME STEP
    C VARIABLES ISPOB(200) AND ISPOR(200) ARE THE ACTUAL SPEEDS
    IF (MBREECH(ICOMMAND,1).EQ.1) SPEED=4.
    IBSA(ICOMMAND,1)=INT(SPEED)
    IF (IBAVL(ICOMMAND,1).EQ.0) IBSA(ICOMMAND,1)=0
    IDEFB(ICOMMAND)=0
  C
  C MTYPEB(ICOMMAND)=4
  C IF (SPEED.LE.0.0) MODEMB(ICOMMAND)=3
  C ELSE
  C IDEFR(ICOMMAND)=0
  C IF (MBREECH(ICOMMAND,2).EQ.1) SPEED=4.
  C IRSA(ICOMMAND,1)=INT(SPEED)
  C IF (IRAVL(ICOMMAND,1).EQ.0) IRSA(ICOMMAND,1)=0
  C MTYPEB(ICOMMAND)=4
  C IF (SPEED.LE.0.0) MODEMR(ICOMMAND)=3
  C ENDF
  C DO 200 L=NSUB1,NSUB2
  C IF (KSIDE.EQ.1) THEN
  C IDEFB(I)=0
  C IF (MBREECH(I,1).EQ.1) SPEED=4.
  C IF (IBSYS(1,2).GE.1) THEN
  C IBSA(I,1)=INT(SPEED)
  C IF (IBAVL(I,1).EQ.0) IBSA(I,1)=0
  C MTYPEB(I)=4
  C IF (SPEED.LE.0.0) MODEMB(I)=3
  C ENDF
  C ELSE
  C IDEFR(I)=0
  C IF (MBREECH(I,2).EQ.1) SPEED=4.
  C IF (IRSYS(1,2).GE.1) THEN
  C IRSA(I,1)=INT(SPEED)
  C IF (IRAVL(I,1).EQ.0) IRSA(I,1)=0
  C MTYPEB(I)=4
  C IF (SPEED.LE.0.0) MODEMR(I)=3
  C ENDF
  C ENDF
  C CONTINUE
  C ENDF
  IF (MTYPE.EQ.5) GO TO 10
  GO TO 150
1000 CONTINUE
RETURN
END
SUBROUTINE FDIR (AZIN,AZOUT)
C
C TAKES AZIMUTH ANGLE IN AND CONVERTS TO CARTESIAN ANGLE OUT
C
  AZOUT=-AZIN+90.
  IF (AZOUT.LT.0.0) AZOUT=AZOUT+360.
  RETURN
  ENDF
SUBROUTINE FAZ (AZOUT,AZIN)
C
C TAKES CARTESIAN ANGLE OUT AND GIVES AZIMUTH ANGLE IN
C
  AZIN=-(AZOUT-90.)
  IF (AZIN.LT.0.0) AZIN=AZIN+360.
  RETURN
  ENDF
SUBROUTINE BATTLE2
  BATTLE.FOR
C
C MOVE AND SHOOT UNITS
C
C CYCLES THROUGH SIXTY MOVEMENT AND SHOOTING TIME STEPS
C THEN RETURNS TO COMMAND LEVEL
C
  COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
  COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),IC(1100),
  & NC(15),NCO(15),JCON
  COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
  COMMON/FORCE/GHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
  & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
  COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
  & L1BM,L2M,L3M,L1RM
  COMMON/WEAP/ONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
  & NRWB(10,3),NRWR(10,3),RFB(180),RR(180),PKB(180,20),PKR(180,20),
  & NRNB(30),NRNR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
  & TEMINR(30),ACQB(10,5),ACQR(10,5),SRCHB(10),SRCHR(10),
  & MILB(10,3),MILR(10,3)
  COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
  COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
  & ,BXYF(200,2),RXYF(200,2),ISPB(200),ISPR(200)
  COMMON/BATTLE/ITIME,ITIME,ITGTR(200,2),ITGTR(200,2),
  & PKTB(200,2),PKTR(200,2),IKLLB(200),IKLLR(200),IAMB(200),
  & IAMB(200)
  COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
  & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),

```

80  
C

**C**

```

C IF (IBAVL(K).EQ.0) THEN
C THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
C SEE IF IT IS TIME TO MAKE IT AVAILABLE AGAIN
  IF (ITIME.GE.ITAVL(K,1)) IBAVL(K)=1
  ENDF
ENDIF
ENDIF

C UPDATE TERRAIN GRADE FOR THIS VEHICLE
KSIDE=1
CALL GRADE (KSIDE,K)

  BACEL(K)=0.0
C COMPARE DESIRED SPEED AGAINST ACTUAL SPEED ON THIS GRADE
C AND DETERMINE CURRENT ACCELERATION
CALL FINDACEL (KSIDE,K,ARATE)
  BACEL(K)=ARATE

C CHECK ACCELERATION AND SELECT MANEUVER MODE
IF (ARATE.EQ.0.0) MODEMB(K)=1
C VEHICLE DRIVES AT CURRENT SPEED
IF (ARATE.LE.0.0.AND.ISPDB(K).EQ.0) THEN
C VEHICLE IS STOPPED AND HAS NO DESIRE TO MOVE
  MODEMB(K)=3
  ARATE=0.0
  ENDF
IF (ARATE.LT.0.0.AND.IBSA(K,1).NE.0) THEN
C DECELERATE ACCORDINGLY
  MODEMB(K)=1
  ENDF
IF (ARATE.LT.0.0.AND.IBSA(K,1).EQ.0) THEN
C STOP VEHICLE
  MODEMB(K)=3
  ISPDB(K)=IBSA(K,1)
  ARATE=0.0
  ENDF
IF (ARATE.GT.0.0) MODEMB(K)=2
C ACCELERATE ACCORDINGLY

C INCREASE CYCLE TIME IN THIS MODE
TCMMB(K,MODEMB(K))=TCMMB(K,MODEMB(K))+DT

C IF (MTYPEB(K).NE.1) THEN
C ALL OF THESE MOVEMENT TYPES SIMPLY INVOLVE CONTINUING IN
C CURRENT DIRECTION UNTIL OBJECTIVE IS REACHED
  AZIN=IBSA(K,2)
  CALL FDIR (AZIN,AZOUT)
  CHECK CURRENT DISTANCE TO OBJECTIVE
  ROB1=((BXYA(K,1)-BXYO(K,1))**2+(BXYA(K,2)-BXYO(K,2))**2)**.5
  MOVE UNIT
  DELPOS=(ISPDB(K)*DT+ARATE/2*DT*DT)/3600.

C before moving, evaluate minefields
  IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
    CALL MINEFIELD (1,K,DELPOS,AZOUT,INNOW,INNEXT)
    IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
      C IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
      INKNOW=0
      NXKNOW=0
      IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
      IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
      IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
        IF (MBREECH(K,1).EQ.0) THEN
          C STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
          DELPOS=0.0
          ARATE=0.0
          ISPDB(K)=0
          MODEMB(K)=3
          IBSA(K,1)=0
          ENDF
          IF (MBREECH(K,1).EQ.-1)
            CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
          ELSE
            C YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
            C OR YOU'RE BREECING IT.
            C EVALUATE PROBABILITY OF BEING KILLED
            IF (MBREECH(K,1).NE.1)
              CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
            ENDF
            ENDF
            ENDF
          C1 BXYA(K,1)=BXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
          BXYA(K,2)=BXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
          C CHANGE SPEED
          ISPDB(K)=ISPDB(K)+ARATE*DT
          C CHECK NEW DISTANCE TO OBJECTIVE
          ROB2=((BXYA(K,1)-BXYO(K,1))**2+(BXYA(K,2)-BXYO(K,2))**2)**.5
          C IF THIS DISTANCE HAS GOTTEN BIGGER, THEN OBJECTIVE HAS JUST
          C BEEN PASSED, STOP VEHICLE
          IF (ROB2.GE.ROB1) THEN
            IBSA(K,1)=0
            ISPDB(K)=0
            MODEMB(K)=3
            ENDF
          ELSE
            C FOLLOW-ME MOVEMENT TYPE, CHECK TO SEE IF PIVOT POINT IS
            C NOW REACHED. IF REACHED, CHANGE MOVEMENT TYPE AND AZIMUTH
            C AND OBJECTIVE
            AZIN=IBSA(K,2)
            CALL FDIR (AZIN,AZOUT)
            CHECK CURRENT DISTANCE TO PIVOT POINT

```

```

  ROB1=((BXYA(K,1)-BXYO(K,1))**2+(BXYA(K,2)-BXYO(K,2))**2)**.5
C MOVE UNIT
  DELPOS=(ISPDB(K)*DT+ARATE/2*DT*DT)/3600.
C1 before moving, evaluate minefields
  IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
    CALL MINEFIELD (1,K,DELPOS,AZOUT,INNOW,INNEXT)
    IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
      C IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
      INKNOW=0
      NXKNOW=0
      IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
      IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
      IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
        IF (MBREECH(K,1).EQ.0) THEN
          C STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
          DELPOS=0.0
          ARATE=0.0
          ISPDB(K)=0
          MODEMB(K)=3
          IBSA(K,1)=0
          ENDF
          IF (MBREECH(K,1).EQ.-1)
            CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
          ELSE
            C YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
            C OR YOU'RE BREECING IT.
            C EVALUATE PROBABILITY OF BEING KILLED
            IF (MBREECH(K,1).NE.1)
              CALL MINEKILL (1,K,INNOW,INNEXT,DELPOS)
            ENDF
            ENDF
            ENDF
          C1 BXYA(K,1)=BXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
          BXYA(K,2)=BXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
          C CHANGE SPEED
          ISPDB(K)=ISPDB(K)+ARATE*DT
          C CHECK NEW DISTANCE TO PIVOT POINT
          ROB2=((BXYA(K,1)-BXYO(K,1))**2+(BXYA(K,2)-BXYO(K,2))**2)**.5
          C IF THIS DISTANCE HAS GOTTEN BIGGER, THEN PIVOT POINT HAS JUST
          C BEEN PASSED, TURN VEHICLE
          IF (ROB2.GE.ROB1) THEN
            MTYPEB(K)=0
            X2=BXYO(K,1)
            Y2=BXYO(K,2)
            X1=BXYA(K,1)
            Y1=BXYA(K,2)
            TOP=Y2-Y1
            BOT=X2-X1
            IF (BOT.EQ.0.0) BOT=BOT+.001
            DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
            IF (BOT.LT.0.0) THEN
              IF (TOP.LT.0.0) DIR=DIR+180.
              IF (TOP.GT.0.0) DIR=180.-DIR
            ELSE
              IF (TOP.LT.0.0) DIR=360.-DIR
            ENDF
            AZOUT=DIR
            CALL FAZ (AZOUT,AZIN)
            IAZ=INT(AZIN)
            IBSA(K,2)=IAZ
            ENDF
            ENDF
          C SET TURRET POSITION IF STILL IN SURVEILLANCE MODE
          IF (MODEFB(K).EQ.7) AZTB(K)=IBSA(K,2)
          130 CONTINUE
          C RED
          DO 140 K=1,NR
          C CHECK TO SEE IF THIS UNIT IS KILLED
          IF (IRSYS(K,2).EQ.0) GO TO 140
          C CHECK TO SEE IF THIS IS AN ARTILLERY UNIT AND SHOULD BEGIN
          C SCOOTING
          KASYS=IRSYS(K,1)
          IF (KASYS.GT.10) THEN
            KASYS=KASYS-10
            IF (ISCOOTR(KASYS,1).EQ.1) THEN
              C THIS UNIT SHOOTS AND SCOOT
              IF (IRAVL(K).EQ.0) THEN
                C THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
                IF (ITIME.GE.ITSC(K,2)) THEN
                  C ITS TIME TO SCOOT
                  IRAVL(K)=1
                  IRSA(K,1)=50
                  C DEFINE NEW MOVEMENT OBJECTIVE BASED ON SCOOT DISTANCE
                  AZIMUTH=REAL(IRSA(K,2))
                  CALL FDIR (AZIMUTH,AZOUT)
                  OBJR=REAL(ISCOTR(KASYS,3))/1000.
                  X1=RXYA(K,1)
                  Y1=RYXA(K,2)
                  X2=X1+OBJR*COS(AZOUT/180.*PI)
                  Y2=Y1+OBJR*SIN(AZOUT/180.*PI)
                  RXYO(K,1)=X2
                  RXYO(K,2)=Y2
                  ENDF
                  ENDF
                  IF (IRAVL(K).EQ.-1) THEN
                    C THIS UNIT IS CURRENTLY SCOOTING
                    C SEE IF IT IS TIME TO UN-SCOOT IT AND MAKE IT AVAILABLE FOR
                    C A FIRE MISSION

```

```

      IF (TIME.GE.ITAVL(K,2)) IRVL(K)=1
      ENDF
    ELSE
C     THIS ARTILLERY UNIT DOES NOT SCOOT
C     SEE IF IT IS INVOLVED IN A FIRE MISSION AND IS NOW TIME TO
C     MAKE AVAILABLE AGAIN
      IF (IRVL(K).EQ.0) THEN
C     THIS UNIT IS CURRENTLY INVOLVED IN A FIRE MISSION
C     SEE IF IT IS TIME TO MAKE IT AVAILABLE AGAIN
      IF (TIME.GE.ITAVL(K,2)) IRVL(K)=1
      ENDF
      ENDF
    ENDIF
C     UPDATE TERRAIN GRADE FOR THIS VEHICLE
      KSIDE=2
      CALL GRADE (KSIDE,K)
C     COMPARE DESIRED SPEED AGAINST ACTUAL SPEED ON THIS GRADE
C     AND DETERMINE CURRENT ACCELERATION
      CALL FINDACEL (KSIDE,K,ARATE)
C     CHECK ACCELERATION AND SELECT MANEUVER MODE
      IF (ARATE.EQ.0.0) MODEMR(K)=1
C     VEHICLE DRIVES AT CURRENT SPEED
      IF (ARATE.LE.0.0.AND.ISPDR(K).EQ.0) THEN
C     VEHICLE IS STOPPED AND HAS NO DESIRE TO MOVE
      MODEMR(K)=3
      ARATE=0.0
      ENDF
      IF (ARATE.LT.0.0.AND.IRSA(K,1).NE.0) THEN
C     DECELERATE ACCORDINGLY
      MODEMR(K)=1
      ENDF
      IF (ARATE.LT.0.0.AND.IRSA(K,1).EQ.0) THEN
C     STOP VEHICLE
      MODEMR(K)=3
      ISPDR(K)=IRSA(K,1)
      ARATE=0.0
      ENDF
      IF (ARATE.GT.0.0) MODEMR(K)=2
C     ACCELERATE ACCORDINGLY
      INCREASE CYCLE TIME IN THIS MODE
      TCMMR(K,MODEMR(K))=TCMMR(K,MODEMR(K))+DT
C     IF (MTYPER(K).NE.1) THEN
C     ALL OF THESE MOVEMENT TYPES SIMPLY INVOLVE CONTINUING IN
C     CURRENT DIRECTION UNTIL OBJECTIVE IS REACHED
      AZIN=IRSA(K,2)
      CALL FDIR (AZIN,AZOUT)
C     CHECK CURRENT DISTANCE TO OBJECTIVE
      ROB1=((RXYA(K,1)-RXYO(K,1))**2+(RXYA(K,2)-RXYO(K,2))**2)**.5
C     MOVE UNIT
      DELPOS=(ISPDR(K)*DT+ARATE/2*DT*DT)/3600.
C     before moving, evaluate minefields
      IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
      CALL MINEFIELD (2,K,DELPOS,AZOUT,INNOW,INNEXT)
      IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
C     IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
      INKNOW=0
      NXKNOW=0
      IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
      IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
      IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
      IF (MBREECH(K,2).EQ.0) THEN
C     STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
      DELPOS=0.0
      ARATE=0.0
      ISPDR(K)=0
      MODEMR(K)=3
      IRSA(K,1)=0
      ENDF
      IF (MBREECH(K,2).EQ.-1)
      CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
      ELSE
C     YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
C     EVALUATE PROBABILITY OF BEING KILLED
      IF (MBREECH(K,2).NE.1)
      CALL MINEKILL (2,K,INNOW,INNEXT,DELPOS)
      ENDF
      ENDF
      ENDF
      IF (MTYPER(K).EQ.7) AZTR(K)=IRSA(K,2)
      CONTINUE
    EXECUTE DIRECT-FIRE ORDERS
    ALL ACTIVITY OCCURS AT THE END OF THE MODE TIME
    THAT WAY WASTED TIME AND WASTED ENGAGEMENTS HAVE
    AN EFFECT ON THE BATTLE
    BLUE
    DO 150 K=1,NB
      LOOP 1
      CHECK TO SEE IF THIS UNIT IS KILLED AND SKIP
      WRITE (*,*) 'KILL STATUS BLUE',K,'ALIVE'
      BUT BEFORE SKIPPING IT, SEE IF IT WAS A FIRE-AND-FORGET
      WEAPON AND IN THE LAST SECONDS OF ENGAGEMENT MODE OR
      RE-ENGAGEMENT MODE.
      MODEFB=MODEFB(K)
      IF (IBSYS(K,2).EQ.0) THEN
      NSYS=IBSYS(K,1)
      IF (MODEFB.EQ.2.OR.MODEFB.EQ.6) THEN
      NWP=ITGT8(K,2)
      MILTYPE=MIL8(NSYS,NWP)
      IF (MILTYPE.EQ.0) THEN
      CALCULATE TIME TO BULLET IMPACT
      TCHANGE=TRMFB(K)
      BORE=BORE8(NSYS,NWP)/1000.
      TOF=TOF8(NSYS,NWP)
      TRNG=PKT8(K,2)
      TTRNG=TRNG/BORE*TOF+1.0
      IF (TIME.GE.(TCHANGE-TTRNG)) THEN
      THIS WEAPON IS NOW IN SACRIFICE MODE SINCE IT IS DEAD AND
      HAPPENED TO GET A FIRE-AND-FORGET SHOT OFF FIRST

```

```

C     AND OBJECTIVE
      AZIN=IRSA(K,2)
      CALL FDIR (AZIN,AZOUT)
C     CHECK CURRENT DISTANCE TO PIVOT POINT
      ROB1=((RXYA(K,1)-RXYP(K,1))**2+(RXYA(K,2)-RXYP(K,2))**2)**.5
C     MOVE UNIT
C     before moving, evaluate minefields
      IF (NMINES.GE.1.AND.DELPOS.GT.0.0) THEN
      CALL MINEFIELD (2,K,DELPOS,AZOUT,INNOW,INNEXT)
      IF (INNOW.NE.0.OR.INNEXT.NE.0) THEN
C     IN OR ENTERING A MINEFIELD, DOES HE KNOW IT?
      INKNOW=0
      NXKNOW=0
      IF (INNOW.NE.0.AND.NAZMINE(INNOW,3).EQ.1) INKNOW=1
      IF (INNEXT.NE.0.AND.NAZMINE(INNEXT,3).EQ.1) NXKNOW=1
      IF (INKNOW.EQ.1.OR.NXKNOW.EQ.1) THEN
      IF (MBREECH(K,2).EQ.0) THEN
C     STOP AND AWAIT ORDERS, YOU'RE IN OR GOING INTO A KNOWN MINEFIELD
      DELPOS=0.0
      ARATE=0.0
      ISPDR(K)=0
      MODEMR(K)=3
      IRSA(K,1)=0
      ENDF
      IF (MBREECH(K,2).EQ.-1)
      CALL MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
      ELSE
C     YOU'RE MOVING THROUGH A MINEFIELD AND DON'T KNOW IT
C     EVALUATE PROBABILITY OF BEING KILLED
      IF (MBREECH(K,2).NE.1)
      CALL MINEKILL (2,K,INNOW,INNEXT,DELPOS)
      ENDF
      ENDF
      ENDF
      DELPOS=(ISPDR(K)*DT+ARATE/2*DT*DT)/3600.
      RXYA(K,1)=RXYA(K,1)+DELPOS*COS(AZOUT*PI/180.)
      RXYA(K,2)=RXYA(K,2)+DELPOS*SIN(AZOUT*PI/180.)
C     CHANGE SPEED
      ISPDR(K)=ISPDR(K)+ARATE*DT
C     CHECK NEW DISTANCE TO PIVOT POINT
      ROB2=((RXYA(K,1)-RXYP(K,1))**2+(RXYA(K,2)-RXYP(K,2))**2)**.5
C     IF THIS DISTANCE HAS GOTTEN BIGGER, THEN PIVOT POINT HAS JUST
C     BEEN PASSED, TURN VEHICLE
      IF (ROB2.GE.ROB1) THEN
      MTYPER(K)=0
      X2=RXYO(K,1)
      Y2=RXYO(K,2)
      X1=RXYA(K,1)
      Y1=RXYA(K,2)
      TOF=Y2-Y1
      BOT=X2-X1
      IF (BOT.EQ.0.0) BOT=BOT+.001
      DIR=ATAN2ABS(TOPVABS(BOT))/180./PI
      IF (BOT.LT.0.0) THEN
      IF (TOP.LT.0.0) DIR=DIR+180.
      IF (TOP.GT.0.0) DIR=180.-DIR
      ELSE
      IF (TOP.LT.0.0) DIR=360.-DIR
      ENDF
      AZOUT=DIR
      CALL FAZ (AZOUT,AZIN)
      IAZ=INT(AZIN)
      IRSA(K,2)=IAZ
      ENDF
      ENDF
C     SET TURRET POSITION IF STILL IN SURVEILLANCE MODE
      IF (MODEFR(K).EQ.7) AZTR(K)=IRSA(K,2)
      CONTINUE
    EXECUTE DIRECT-FIRE ORDERS
    ALL ACTIVITY OCCURS AT THE END OF THE MODE TIME
    THAT WAY WASTED TIME AND WASTED ENGAGEMENTS HAVE
    AN EFFECT ON THE BATTLE
    BLUE
    DO 150 K=1,NB
      LOOP 1
      CHECK TO SEE IF THIS UNIT IS KILLED AND SKIP
      WRITE (*,*) 'KILL STATUS BLUE',K,'ALIVE'
      BUT BEFORE SKIPPING IT, SEE IF IT WAS A FIRE-AND-FORGET
      WEAPON AND IN THE LAST SECONDS OF ENGAGEMENT MODE OR
      RE-ENGAGEMENT MODE.
      MODEFB=MODEFB(K)
      IF (IBSYS(K,2).EQ.0) THEN
      NSYS=IBSYS(K,1)
      IF (MODEFB.EQ.2.OR.MODEFB.EQ.6) THEN
      NWP=ITGT8(K,2)
      MILTYPE=MIL8(NSYS,NWP)
      IF (MILTYPE.EQ.0) THEN
      CALCULATE TIME TO BULLET IMPACT
      TCHANGE=TRMFB(K)
      BORE=BORE8(NSYS,NWP)/1000.
      TOF=TOF8(NSYS,NWP)
      TRNG=PKT8(K,2)
      TTRNG=TRNG/BORE*TOF+1.0
      IF (TIME.GE.(TCHANGE-TTRNG)) THEN
      THIS WEAPON IS NOW IN SACRIFICE MODE SINCE IT IS DEAD AND
      HAPPENED TO GET A FIRE-AND-FORGET SHOT OFF FIRST

```



```

MODEF=0
MODEFB(K)=MODEF
NSHOTB(K,NWP)=NSHOTB(K,NWP)-1
CALL OUTPUT2 (TIME,1,K,0)
C WRITE (",") BLUE 'K' MAKES SACRIFICE SHOT
WRITE (10,") BLUE 'K' MAKES SACRIFICE SHOT
ELSE
MODEFB(K)=0
C FIRE MODE SET TO 0 - KILLED WHILE ENGAGING A TARGET
ENDIF
ENDIF
ENDIF
ENDIF

IF (IBSYS(K,2).EQ.0.AND.MODEF.NE.0) GO TO 150

C IF (MODEF.EQ.0) THEN
C RE-EVALUATE RE-ENGAGEMENT MODE TO SEE IF THIS RED TARGET
C WAS JUST KILLED
IF (IRSYS(ITGTB(K,1),2).EQ.1) THEN
MODEF=0
MODEFB(K)=0
WRITE (10,") BLUE 'K' RE-ENGAGES RED 'ITGTB(K,1)
ELSE
C THIS TARGET WAS KILLED LAST GO AROUND SO FIND ANOTHER
MODEF=1
MODEFB(K)=1
NSYS=IBSYS(K,1)
ALFT1=IBPRIME(K,1)
ARGT1=IBPRIME(K,2)
IF (ALFT1.GT.ARG1) ARGT1=ARGT1+360.
ALFT2=IBSECOND(K,1)
ARGT2=IBSECOND(K,2)
IF (ALFT2.GT.ARG2) ARGT2=ARGT2+360.
IF (IBPRIME(K,1).EQ.IBSECOND(K,1).AND.IBPRIME(K,2).
EQ.IBSECOND(K,2)) THEN
ALFT2=-1.0
ARGT2=-1.0
ENDIF
C DETERMINE END TIME OF SEARCH MODE CYCLE
STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2)/SRCHB(NSYS)
TRMFB(K)=TIME+STIME
ENDIF
ENDIF

C WRITE (",") FIRE MODE BLUE 'MODEF
C INCREASE CYCLE TIME IN THIS MODE
TCFMB(K,MODEF)=TCFMB(K,MODEF)+DT
C WRITE (",") CYCLE TIME BLUE 'TCFMB(K,MODEF)

IF (MODEF.EQ.1.OR.MODEF.EQ.4.OR.MODEF.EQ.5.OR.MODEF.EQ.3) THEN
LOOP 2

SEARCH MODE (1) OR BALK MODE (4) OR WAIT MODE (5)
EM ENERGY WAIT MODE (3)
BALK MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
FOR DUTY CYCLE ACCOUNTABILITY
WAIT MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
FOR DUTY CYCLE ACCOUNTABILITY
EM ENERGY WAIT MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
FOR DUTY CYCLE ACCOUNTABILITY

CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
TCHANGE=TRMFB(K)
C WRITE (",") FIRE MODE CHANGE BLUE 'TCHANGE
TNEXT=TIME+DTIME
IF (TCHANGE.LE.TNEXT) THEN
LOOP 3
IT'S TIME TO FIND A TARGET

FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
SO THAT MASKING CAN BE EVALUATED

KSIDE=1
MYSIDE=1
BLUE=K
CALL VMASK (KSIDE,BLUE)

C FIND ALL RED TARGETS IN PRIMARY AND SECONDARY ZONES
C OUT TO MAX RANGE FOR EACH ON-BOARD WEAPON
NSYS=IBSYS(K,1)
NWEAP=NWB(NSYS)
NRNG1=NRWB(NSYS,1)
RMAX1=0.0
RMAX2=0.0
RMAX3=0.0
NUM1=0
NUM2=0
NUM3=0
C ISR=(1-J)*18+(J-1)*6+K
C WHERE J=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C K=WEAPON RANGE (8 MAX); N=TARGET (2*10 MAX)
NUM1=(NSYS-1)*18+(1-J)*6+NRNG1
RMAX1=RB(NUM1)
C WRITE (",") RMAX1,RMAX1,RB(22),RB(22)
C WRITE (",") NSYS,NSYS,NWEAP,NWEAP,NRNG1,NRNG1,NUM1,NUM1
C WRITE (",") NSHOTB,NSHOTB(K,1)
C IF (NSHOTB(K,1).LE.0) THEN
C RMAX1=0.0
C NUM1=0
C ENDF
IF (NWEAP.GE.2) THEN

```

```

NRNG2=NRWB(NSYS,2)
NUM2=(NSYS-1)*18+(2-J)*6+NRNG2
RMAX2=RB(NUM2)
C WRITE (",") NSYS,NSYS,NWEAP,NWEAP,NRNG2,NRNG2,NUM2,NUM2
C WRITE (",") NSHOTB,NSHOTB(K,2)
C IF (NSHOTB(K,2).LE.0) THEN
C RMAX2=0.0
C NUM2=0
C ENDF
ENDIF
IF (NWEAP.EQ.3) THEN
NRNG3=NRWB(NSYS,3)
NUM3=(NSYS-1)*18+(3-J)*6+NRNG3
RMAX3=RB(NUM3)
C WRITE (",") NSYS,NSYS,NWEAP,NWEAP,NRNG3,NRNG3,NUM3,NUM3
C WRITE (",") NSHOTB,NSHOTB(K,3)
C IF (NSHOTB(K,3).LE.0) THEN
C RMAX3=0.0
C NUM3=0
C ENDF
ENDIF
ENDIF

C SEARCH ALL ZONES BASED ON ZONE AZIMUTHS
C IOVER1=0
C IOVER2=0
ALFT1=IBPRIME(K,1)
ARGT1=IBPRIME(K,2)
IF (ALFT1.GT.ARG1) IOVER1=1
IF (ALFT1.GT.ARG1) ARGT1=ARGT1+360.
ALFT2=IBSECOND(K,1)
ARGT2=IBSECOND(K,2)
IF (ALFT2.GT.ARG2) IOVER2=1
IF (ALFT2.GT.ARG2) ARGT2=ARGT2+360.
IF (IBPRIME(K,1).EQ.IBSECOND(K,1).AND.IBPRIME(K,2).
EQ.IBSECOND(K,2)) THEN
ALFT2=-1.0
ARGT2=-1.0
ENDIF
C GET XY COORDINATES FOR THIS BLUE UNIT
XB=BXVA(K,1)
YB=BYVA(K,2)
C INITIALIZE PRIORITIES
NTGT1=0
PRI=0.0
MYWP1=0
NTGT2=0
SEC=0.0
MYWP2=0
NTGT3=0
THIRD=0.0
MYWP3=0
TRNG1=0.0
TRNG2=0.0
TRNG3=0.0
PKWP1=0.0
PKWP2=0.0
PKWP3=0.0

C IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C EVALUATED
ITRK1=ITRKB(NSYS,1)
ISROF=(NSYS-1)*3+1
IRNG11=1
IRNG12=1
IRNG13=1
IRNG11=IRNGB(ISROF,1)
IF (NWEAP.GE.2) IRNG12=IRNGB((ISROF+1),1)
IF (NWEAP.GT.2) IRNG13=IRNGB((ISROF+2),1)
ICANRD=1
IF (IRNG11.EQ.0.OR.IRNG12.EQ.0.OR.IRNG13.EQ.0) ICANRD=0
IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
C CALL DMASK (XB,YB,RMAX1,RMAX2,RMAX3)

C FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
C MASK VISUAL AND IR SYSTEMS
C CALL COOKMASK (XB,YB,RMAX1,RMAX2,RMAX3)

C IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C CAN BE EVALUATED
ITRK2=ITRKB(NSYS,2)
ISROF=(NSYS-1)*3+1
IRNG21=1
IRNG22=1
IRNG23=1
IRNG21=IRNGB(ISROF,2)
IF (NWEAP.GE.2) IRNG22=IRNGB((ISROF+1),2)
IF (NWEAP.GT.2) IRNG23=IRNGB((ISROF+2),2)
ICANRE=1
IF (IRNG21.EQ.0.OR.IRNG22.EQ.0.OR.IRNG23.EQ.0) ICANRE=0
IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
C CALL ESMASK (XB,YB,RMAX1,RMAX2,RMAX3)

C IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C CAN BE EVALUATED
ITRK3=ITRKB(NSYS,3)
ISROF=(NSYS-1)*3+1

```

```

C   IRNG31=1
C   IRNG32=1
C   IRNG33=1
C   IRNG31=IRNG31/ISROF.3
C   IF (NWEAP.GE.2) IRNG32=IRNG32/((ISROF+1).3)
C   IF (NWEAP.GT.2) IRNG33=IRNG33/((ISROF+2).3)
C   ICANRS=1
C   IF (IRNG31.EQ.0.OR.IRNG32.EQ.0.OR.IRNG33.EQ.0) ICANRS=0
C   IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
C   & CALL GRMASK (XB,YB,RMAX1,RMAX2,RMAX3)

C   IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C   TRACKING OR RANGING,
C   FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
C   MASKING
C   CAN BE EVALUATED
C   ITRK4=ITRK3/NSYS.4
C   ISROF=(NSYS-1)*3+1
C   IRNG41=1
C   IRNG42=1
C   IRNG43=1
C   IRNG41=IRNG41/ISROF.4
C   IF (NWEAP.GE.2) IRNG42=IRNG42/((ISROF+1).4)
C   IF (NWEAP.GT.2) IRNG43=IRNG43/((ISROF+2).4)
C   ICANRA=1
C   IF (IRNG41.EQ.0.OR.IRNG42.EQ.0.OR.IRNG43.EQ.0) ICANRA=0
C   IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
C   & CALL ASMASK (XB,YB,RMAX1,RMAX2,RMAX3)

C   IFIND=0
C   DO 140 J=1,NR
C   LOOP 4
C   CHECK TO SEE IF THIS RED UNIT IS KILLED
C   IF (RSYS(J).2.EQ.0.OR.RSYS(J).1.GT.10) GO TO 140
C   FIND THIS RED UNIT'S X,Y COORDINATES
C   XR=RXV(J)
C   YR=RYV(J)
C   FIND RANGE TO THIS TARGET
C   RANGE=((XR-XB)**2+(YR-YB)**2)**.5
C   WRITE (":") RANGE TO RED ":J",RANGE

C   CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C   NOT DETECTED
C   CALL CHECKSIG (1,K,J,RANGE,JSKIP)
C   IF (JSKIP.EQ.0) GO TO 140

C   IF THIS TARGET IS OUTSIDE THE RANGE OF ALL WEAPONS SKIP IT
C   WRITE (":") RANGE,RANGE,RMAX1,RMAX1,RMAX2,RMAX2,
C   &RMAX3,RMAX3
C   JSKIP=0
C   IF (RANGE*1000./LE.RMAX1) JSKIP=1
C   IF (RANGE*1000./LE.RMAX2) JSKIP=1
C   IF (RANGE*1000./LE.RMAX3) JSKIP=1
C   WRITE (":") RANGE,JSKIP,"YES"
C   IF (JSKIP.EQ.0) GO TO 140
C   TARGET IS WITHIN RANGE OF ONE OF THESE WEAPONS
C   IF THIS TARGET IS OUTSIDE THE FIRE ORDERS RANGE WINDOW SKIP IT
C   IF (RANGE.LT.WRNGB(K.1).OR.RANGE.GT.WRNGB(K.2)) JSKIP=0
C   IF (JSKIP.EQ.0) GO TO 140
C   FIND ANGLE TO THIS TARGET
C   TOP=YR-YB
C   BOT=XR-XB
C   IF (BOT.EQ.0.0) BOT=BOT+.001
C   DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
C   IF (BOT.LT.0.0) THEN
C   IF (TOP.LT.0.0) DIR=DIR-180.
C   IF (TOP.GT.0.0) DIR=180.-DIR
C   ELSE
C   IF (TOP.LT.0.0) DIR=360.-DIR
C   ENDIF
C   CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C   CALL FAZ (DIR,AZIN)

C   AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C   MASKING AND OBSCURANTS AND DROP THIS
C   TARGET FROM ZONE IF UNOBSERVABLE

C   TERRAIN MASKING
C   IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
C   IF (JSKIP.EQ.0) GO TO 140

C   VEHICLE MASKING
C   CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)

C   BURNING VEHICLE MASKING
C   CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)
C   IF (JSKIP.EQ.0) GO TO 140

C   DUST MASKING
C   IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C   TRACKING OR RANGING,
C   EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
C   JDUST=1
C   IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
C   & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C   IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE THROUGH IT
C   SKIP THIS TARGET
C   IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 140
C   THERE MAY BE DUST IN THE WAY, BUT BLUE CAN SEE THROUGH IT.
C   THE QUESTION REMAINS WHETHER BLUE CAN RANGE OR TRACK THROUGH

```

```

C   THE DUST. IF JDUST=0 THERE IS DUST IN THE WAY.

C   ENGINE SMOKE MASKING
C   IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C   TRACKING OR RANGING,
C   EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C   JENG=1
C   IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
C   & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
C   IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 140

C   SMOKE GRENADE CURTAIN MASKING
C   IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C   TRACKING OR RANGING,
C   EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
C   JSMK=1
C   IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
C   & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
C   IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 140

C   ARTILLERY SMOKE MASKING
C   IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION,
C   TRACKING OR RANGING,
C   EVALUATE ALL ARTILLERY SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C   JART=1
C   IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
C   & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
C   IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 140

C   INZONE=3
C   IF (AZIN.GE.ALFT1.AND.AZIN.LE.ARG1) INZONE=1
C   IF (OVER1.EQ.1.AND.AZIN.LE.(ARG1-360.0)) INZONE=1
C   TARGET IS IN PRIMARY ZONE
C   IF (AZIN.GE.ALFT2.AND.AZIN.LE.ARG2) INZONE=2
C   IF (OVER2.EQ.1.AND.AZIN.LE.(ARG2-360.0)) INZONE=2
C   TARGET IS IN SECONDARY ZONE

C   WRITE (":") RED ":J",INZONE ":INZONE
C   IF (INZONE.GE.1) THEN
C   LOOP 5
C   GIVE THIS TARGET A PRIORITY NUMBER
C   AND IN THE END SELECT ONLY THE HIGHEST PRIORITY TARGET
C   TARGET PRIORITY IS A COMPOSITE OF THE PROBABILITY OF THIS
C   TARGET KILLING YOU TIMES YOUR PROBABILITY OF KILLING IT.
C   NITSYS=RSYS(J)
C   CYCLE THROUGH ALL WEAPONS ON YOUR SYSTEM FOR THE ONE WITH THE
C   HIGHEST PK AGAINST THIS TARGET AND EVALUATE WHETHER YOU WILL
C   SHOOT IT BASED ON NUMBER OF AVAILABLE ROUNDS OF AMMUNITION
C   THE PK AGAINST THIS TARGET, AND THE GENERAL THREAT PICTURE
C   PKIT1=0.0
C   PKIT2=0.0
C   PKIT3=0.0
C   KSIDE=1
C   PKITM=0.0
C   WEAP=0

C   IFIND=IFIND+1
C   IF (IFIND.EQ.1) THEN

C   BLUE HAS FOUND ITS FIRST POTENTIAL TARGET

C   EVALUATE EM ENERGY AVAILABILITY FOR THIS BLUE SYSTEM
C   (POTENTIAL MODE 3)

C   IF (NSYS.EQ.NBLUE) THEN
C   EL1=1.0
C   EL2=1.0
C   EL3=1.0
C   ILOST1=0
C   ILOST2=0
C   ILOST3=0
C   IF (NEMW.GE.1) THEN
C   EL1=ELSYS(KVEWP(1,1))
C   IF (EL1.LT.EWP(1,2)) EL1=0.0
C   IF (EL1.LT.1.0) THEN
C   WRITE (10,"") BLUE "K" WEAPON "1" INSUFFICIENT ENERGY
C   &FOR FULL EM SHOT
C   ELSE
C   EL1=1.0
C   ENDIF
C   ENDIF
C   IF (NEMW.GE.2) THEN
C   EL2=ELSYS(KVEWP(2,1))
C   IF (EL2.LT.EWP(2,2)) EL2=0.0
C   IF (EL2.LT.1.0) THEN
C   WRITE (10,"") BLUE "K" WEAPON "2" INSUFFICIENT ENERGY
C   &FOR FULL EM SHOT
C   ELSE
C   EL2=1.0
C   ENDIF
C   ENDIF
C   IF (NEMW.GE.3) THEN
C   EL3=ELSYS(KVEWP(3,1))
C   IF (EL3.LT.EWP(3,2)) EL3=0.0
C   IF (EL3.LT.1.0) THEN
C   WRITE (10,"") BLUE "K" WEAPON "3" INSUFFICIENT ENERGY
C   &FOR FULL EM SHOT
C   ELSE
C   EL3=1.0
C   ENDIF
C   ENDIF

```

```

ENDIF
ENDIF
C END OF EM ENERGY EVALUATION
DO 141 NUM=1,3
C LOOP 6
IF (NUM.EQ.1.AND.NUM1.EQ.0) GO TO 141
IF (NUM.EQ.2.AND.NUM2.EQ.0) GO TO 141
IF (NUM.EQ.3.AND.NUM3.EQ.0) GO TO 141
IF (NUM.EQ.1) THEN
CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG1,RANGE,PKIT1,J)
IF (ISPD8(K).GT.0.OR.ISPDR(J).GT.0) THEN
MILTYPE=MILB(NSYS,NUM)
IF (MILTYPE.EQ.0) THEN
BORE=BOREB(NSYS,NUM)
TOF=TOFB(NSYS,NUM)
CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
ENDIF
ENDIF
IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
EVALUATE DUST EFFECTS ON WEAPON RANGING AND TRACKING
C IF THIS WEAPON IS F-A-F THEN DUST DEGRADES HIT PROBABILITY
C IF THIS WEAPON CANNOT RANGE THROUGH THE DUST
C IF THIS WEAPON IS M-L THEN DUST DEGRADES TRACKING
C IF THIS WEAPON CANNOT TRACK THROUGH THE DUST
IF (IRNG11.EQ.0.OR.IRNG21.EQ.0) THEN
THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C EVALUATE M-L OR F-A-F
C MILTYPE=MILB(NSYS,NUM)
IF (MILTYPE.EQ.0) THEN
THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
BORE=BOREB(NSYS,NUM)
TOF=TOFB(NSYS,NUM)
CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
ELSE
SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
PKIT1=0.0
ENDIF
ENDIF
NSHOT3=NSHOTB(K,NUM)
C WRITE ('. ') LOG BOT1,LOG(1.-PKIT1),PKIT1,PKIT1
IF (PKIT1.NE.0.0) THEN
IF (NSYS.EQ.NBLUE) PKIT1=PKIT1*EL1
IF (PKIT1.NE.0.0) THEN
SKILL=LOG(.01)/LOG(1.-PKIT1)
ELSE
SKILL=9999.
LOST1=1
ENDIF
ELSE
SKILL=9999.
LOST1=1
ENDIF
ENDIF
IF (NSHOT3.EQ.0) PKIT1=0.0
IF (SKILL.GT.16.0) PKIT1=0.0
IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
IF (SKILL.GT.7.0) THEN
PSHOOT=NSHOT3/SKILL
PNO=RND()
IF (PNO.GE.PSHOOT) PKIT1=0.0
ENDIF
IF (PKIT1.GT.PKITM) THEN
PKITM=PKIT1
IWEAP=1
ENDIF
ENDIF
IF (NUM.EQ.2) THEN
CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG2,RANGE,PKIT2,J)
IF (ISPD8(K).GT.0.OR.ISPDR(J).GT.0) THEN
MILTYPE=MILB(NSYS,NUM)
IF (MILTYPE.EQ.0) THEN
BORE=BOREB(NSYS,NUM)
TOF=TOFB(NSYS,NUM)
CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
ENDIF
ENDIF
IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
IF (IRNG12.EQ.0.OR.IRNG22.EQ.0) THEN
THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C EVALUATE M-L OR F-A-F
C MILTYPE=MILB(NSYS,NUM)
IF (MILTYPE.EQ.0) THEN
THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
BORE=BOREB(NSYS,NUM)
TOF=TOFB(NSYS,NUM)
CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
ELSE
SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
PKIT2=0.0
ENDIF
ENDIF
NSHOT3=NSHOTB(K,NUM)
C WRITE ('. ') LOG BOT2,LOG(1.-PKIT2),PKIT2,PKIT2
IF (PKIT2.NE.0.0) THEN
IF (NSYS.EQ.NBLUE) PKIT2=PKIT2*EL2
IF (PKIT2.NE.0.0) THEN
SKILL=LOG(.01)/LOG(1.-PKIT2)
ELSE
SKILL=9999.
LOST2=1

```

```

ENDIF
ELSE
SKILL=9999.
ENDIF
IF (NSHOT3.EQ.0) PKIT2=0.0
IF (SKILL.GT.16.0) PKIT2=0.0
IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
IF (SKILL.GT.7.0) THEN
PSHOOT=NSHOT3/SKILL
PNO=RND()
IF (PNO.GE.PSHOOT) PKIT2=0.0
ENDIF
IF (PKIT2.GT.PKITM) THEN
PKITM=PKIT2
IWEAP=2
ENDIF
ENDIF
IF (NUM.EQ.3) THEN
CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG3,RANGE,PKIT3,J)
IF (ISPD8(K).GT.0.OR.ISPDR(J).GT.0) THEN
MILTYPE=MILB(NSYS,NUM)
IF (MILTYPE.EQ.0) THEN
BORE=BOREB(NSYS,NUM)
TOF=TOFB(NSYS,NUM)
CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
ENDIF
ENDIF
IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
IF (IRNG13.EQ.0.OR.IRNG23.EQ.0) THEN
THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
C EVALUATE M-L OR F-A-F
C MILTYPE=MILB(NSYS,NUM)
IF (MILTYPE.EQ.0) THEN
THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
BORE=BOREB(NSYS,NUM)
TOF=TOFB(NSYS,NUM)
CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
ELSE
SKIP THIS WEAPON SINCE CAN'T TRACK THE TARGET
PKIT3=0.0
ENDIF
ENDIF
NSHOT3=NSHOTB(K,NUM)
C WRITE ('. ') LOG BOT3,LOG(1.-PKIT3),PKIT3,PKIT3
IF (PKIT3.NE.0.0) THEN
IF (NSYS.EQ.NBLUE) PKIT3=PKIT3*EL3
IF (PKIT3.NE.0.0) THEN
SKILL=LOG(.01)/LOG(1.-PKIT3)
ELSE
SKILL=9999.
LOST3=1
ENDIF
ELSE
SKILL=9999.
LOST3=1
ENDIF
ENDIF
IF (NSHOT3.EQ.0) PKIT3=0.0
IF (SKILL.GT.16.0) PKIT3=0.0
IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
IF (SKILL.GT.7.0) THEN
PSHOOT=NSHOT3/SKILL
PNO=RND()
IF (PNO.GE.PSHOOT) PKIT3=0.0
ENDIF
IF (PKIT3.GT.PKITM) THEN
PKITM=PKIT3
IWEAP=3
ENDIF
ENDIF
141 CONTINUE
C END OF BLUE WEAPON SELECTION LOOP
C WRITE ('. ') 'BLUE WEAPON SELECTION',IWEAP,PK,PKITM
C IF TARGET IS AT A QUESTIONABLE EFFECTIVE RANGE SKIP IT
IF (PKITM.LE.0.0) GO TO 143
IF (IWEAP.EQ.0.0) GO TO 143
C FIND PROBABILITY OF THIS TARGET KILLING YOU AND APPLY PRIORITY FACTOR
C KPFAC=KPRIS(K,NITSYS)
PKYOU1=0.0
PKYOU2=0.0
PKYOU3=0.0
KSIDE=2
NUMIT=NWR(NITSYS)
IRNG1=NRWR(NITSYS,1)
IF (NUMIT.GE.2) IRNG2=NRWR(NITSYS,2)
IF (NUMIT.EQ.3) IRNG3=NRWR(NITSYS,3)
DO 142 NUM=1,NUMIT
IF (NUM.EQ.1) CALL
* FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG1,RANGE,PKYOU1,J)
IF (NUM.EQ.2) CALL
* FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG2,RANGE,PKYOU2,J)
IF (NUM.EQ.3) CALL
* FINDPK (KSIDE,NITSYS,NSYS,NUM,IRNG3,RANGE,PKYOU3,J)
142 CONTINUE
C FIND MAXIMUM PK AGAINST YOU
PKYMAX=0.0

```

```

IF (PKYOU1.GT.PKMAX) PKMAX=PKYOU1
IF (PKYOU2.GT.PKMAX) PKMAX=PKYOU2
IF (PKYOU3.GT.PKMAX) PKMAX=PKYOU3

C IF TARGET'S PK ON YOU IS ZERO, GIVE IT A SMALL VALUE SO THAT
C YOU AT LEAST HAVE A TARGET TO SHOOT AT IF NO OTHERS ARE BETTER
C AFTER ALL THIS TROUBLE OF DECIDING IF YOU CAN KILL IT

IF (PKMAX.LE.0.0) PKMAX=0.01

C FIND COMPOSITE KILL PRIORITY

C WRITE ('.') 'PK IT ON YOU,PKMAX

PRIOR=PKITM*PKMAX*KPFACT
ADD A SMALL RANDOM NUMBER SO NO TWO ARE EVER THE SAME
SMALL=RND()
SMALL=SMALL/25
PRIOR=PRIOR+SMALL

C PRIORITIZE THIS TARGET WITHIN PRIMARY OR SECONDARY ZONE
C
C IF (INZONE.EQ.1) THEN
C THIS TARGET IS IN PRIMARY ZONE
IF (PRIOR.GT.PRI) THEN
PRI=PRIOR
NTGT1=J
MYWP1=IWEAP
TRNG1=RANGE
IF (IWEAP.EQ.1) PKWP1=PKIT1
IF (IWEAP.EQ.2) PKWP1=PKIT2
IF (IWEAP.EQ.3) PKWP1=PKIT3
ENDIF
ENDIF
C IF (INZONE.EQ.2) THEN
C THIS TARGET IS IN SECONDARY ZONE
IF (PRIOR.GT.SEC) THEN
SEC=PRIOR
NTGT2=J
MYWP2=IWEAP
TRNG2=RANGE
IF (IWEAP.EQ.1) PKWP2=PKIT1
IF (IWEAP.EQ.2) PKWP2=PKIT2
IF (IWEAP.EQ.3) PKWP2=PKIT3
ENDIF
ENDIF
C IF (INZONE.EQ.3) THEN
C THIS TARGET IS IN 360 DEGREE ZONE
IF (PRIOR.GT.THIRD) THEN
THIRD=PRIOR
NTGT3=J
MYWP3=IWEAP
TRNG3=RANGE
IF (IWEAP.EQ.1) PKWP3=PKIT1
IF (IWEAP.EQ.2) PKWP3=PKIT2
IF (IWEAP.EQ.3) PKWP3=PKIT3
ENDIF
ENDIF
143 CONTINUE
C TARGET WAS SKIPPED TO LINE 143 SINCE IT WAS AT A QUESTIONABLE
C EFFECTIVE RANGE OR INSUFFICIENT AMMO TO RISK MISSING THE SHOT
C WRITE ('.') 'PRI,PRI,NTGT1,NTGT1
C
C ENDIF
C END OF INZONE
C
149 CONTINUE
C END LOOP 4
C END OF FINDING A TARGET FOR THIS BLUE SYSTEM
C
C IF THE SECONDARY TARGET HAS 125% OF THE PRIORITY
C OF THE PRIMARY TARGET, SHOOT AT THE SECONDARY TARGET
ITGTB(K,1)=0
ITGTB(K,2)=0
PKTB(K,1)=0.0
PKTB(K,2)=0.0
TRNG=0.0
MYWP=0
IF (SEC.GE.(1.25*PRI)) THEN
C SHOOT AT TARGET IN SECONDARY ZONE SINCE IT IS COMPELLING
ITGTB(K,1)=NTGT2
ITGTB(K,2)=MYWP2
PKTB(K,1)=PKWP2
PKTB(K,2)=TRNG2
TRNG=TRNG2
MYWP=MYWP2
ELSE
C STAY WITH PRIMARY ZONE TARGET
ITGTB(K,1)=NTGT1
ITGTB(K,2)=MYWP1
PKTB(K,1)=PKWP1
PKTB(K,2)=TRNG1
TRNG=TRNG1
MYWP=MYWP1
ENDIF
C IF THERE ARE NO TARGETS IN PRIMARY AND SECONDARY,
C SHOOT AT THIRD ZONE TARGET (360 DEGREES)
IN3=0
IF (NTGT1.EQ.0.AND.NTGT2.EQ.0) THEN

```

```

ITGTB(K,1)=NTGT3
ITGTB(K,2)=MYWP3
PKTB(K,1)=PKWP3
PKTB(K,2)=TRNG3
TRNG=TRNG3
MYWP=MYWP3
IF (NTGT3.NE.0) IN3=1
ENDIF

C FOR BATTLE MANAGEMENT, CHECK TO SEE IF MORE THAN 4 BLUE SYSTEMS
C ARE ALREADY SERVICING THIS TARGET, IF SO, SKIP IT.
C
IF (IAMR(ITGTB(K,1)).EQ.4) THEN
ITGTB(K,1)=0
ITGTB(K,2)=0
PKTB(K,1)=0.0
PKTB(K,2)=0.0
TRNG=0.0
MYWP=0
ELSE
IAMR(ITGTB(K,1))=IAMR(ITGTB(K,1))+1
ENDIF
C
C WRITE ('.') 'PRIMARY ZONE TARGET,ITGTB(K,1)
C
C EVALUATE TARGET SEARCH RESULTS AND
C CHANGE MODE FOR THIS BLUE SYSTEM ACCORDINGLY
STIME=0.0
MODEN=0
IF (ITGTB(K,1).EQ.0) THEN
C LOOP 7
C NO TARGETS, WAIT MODES (EM ENERGY OR SIMPLY NO TARGETS)
ILOSTEM=0
IF (ILOST1.EQ.1.OR.ILOST2.EQ.1.OR.ILOST3.EQ.1) ILOSTEM=1
IF (ILOSTEM.EQ.1.AND.NSYS.EQ.NBLUE) THEN
C A TARGET WAS LOST DUE TO INSUFFICIENT EM ENERGY ALL AROUND
C GO TO EM ENERGY WAIT MODE (MODE 3)
WRITE (10,') 'BLUE 'K,' INSUFFICIENT ENERGY TO FIRE ANY WPN
C WAIT A COMPLETE SEARCH CYCLE TIME FOR DUTY CYCLE ACCOUNTABILITY
C BUT CREW IS STILL SEARCHING DURING THIS TIME
MODEN=3
MODEFB(K)=MODEN
C DETERMINE END TIME OF ENERGY WAIT/SEARCH MODE CYCLE
STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCH8(NSYS)
TRMFB(K)=ITIME+IDTIME+STIME
ELSE
C SIMPLY NO TARGETS WITHIN EFFECTIVE RANGE
C WAIT A COMPLETE SEARCH CYCLE TIME FOR DUTY CYCLE ACCOUNTABILITY
C BUT CREW IS STILL SEARCHING DURING THIS TIME
MODEN=5
MODEFB(K)=MODEN
C DETERMINE END TIME OF SEARCH MODE CYCLE
STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCH8(NSYS)
TRMFB(K)=ITIME+IDTIME+STIME
C WRITE ('.') 'BLUE 'K,' NO TARGETS '
ENDIF
ELSE
C THERE IS A TARGET WITHIN EFFECTIVE RANGE
C
C SET TURRET AZIMUTH
CALL TURAZ (1,K,ITGTB(K,1),AZTB(K))
C
C SINCE A TARGET WAS FOUND, CREW GOES INTO QUICK SEARCH
C MODE FOR NEXT SEARCH CYCLE, FOCUSING INITIALLY ON TARGETS
C WITHIN A .5 SECOND TRAVERSE FROM THIS TARGET.
C RESET PRIMARY SEARCH ZONE AND SET SECONDARY TO 0.0
NSYS=IBSYS(K,1)
IBSECOND(K,1)=0
IBSECOND(K,2)=0
ALFT1=AZTB(K)-TRATEB(NSYS)/2.0
ARGT1=AZTB(K)+TRATEB(NSYS)/2.0
IF (ALFT1.LT.0.0) ALFT1=ALFT1+360.0
IF (ARGT1.GT.360.0) ARGT1=ARGT1-360.0
IBPRIME(K,1)=ALFT1
IBPRIME(K,2)=ARGT1
ALFT2=0.0
ARGT2=0.0
IF (ALFT1.GT.ARGT1) ARGT1=ARGT1+360.0
IF (ALFT2.GT.ARGT2) ARGT2=ARGT2+360.0
C
C CHECK MODES FOR ENGAGEMENT, AND BALKING
BALK=RND()
IF (BALK.GT.0.975) THEN
C BALKING, WAIT SEARCH TIME, BUT STILL SEARCH
MODEN=4
MODEFB(K)=MODEN
C DETERMINE END TIME OF BALKING MODE CYCLE
STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCH8(NSYS)
TRMFB(K)=ITIME+IDTIME+STIME
131 FORMAT ('BLUE '3,' BALKS AT RED '3)
C WRITE ('.131) K,ITGTB(K,1)
C WRITE (10,131) K,ITGTB(K,1)
ENDIF

```



```

C CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (OPLAZIN)
C
JDUST=1
IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
  & CALL DBEE (XB,YB,XR,YRAZIN,JDUST)
C
C IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE
C TRACK THROUGH IT, THE TARGET IS MISSED
C IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0
C
C IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT BLUE CAN SEE
C THROUGH IT. THE QUESTION REMAINS WHETHER BLUE CAN TRACK THROUGH
C THE DUST.
C
IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0
C
C IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM
C
C BURNING VEHICLE MASKING
CALL COOKSEE (XB,YB,XR,YRAZIN,JSKIP)
C
C ENGINE SMOKE MASKING
C RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
JENG=1
IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
  & CALL ESSEE (XB,YB,XR,YRAZIN,JENG)
IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0
C
C SMOKE GRENADE CURTAIN MASKING
C RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
JSMK=1
IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
  & CALL GRSEE (XB,YB,XR,YRAZIN,JSMK)
IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0
C
C ARTILLERY SMOKE CURTAIN MASKING
C RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
C TARGET
JART=1
IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
  & CALL ASSEE (XB,YB,XR,YRAZIN,JART)
IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0
C
ENDIF
C END OF OBSCURANT RE-EVALUATION FOR M-I-L WEAPON
C
IF (JSKIP.EQ.1) THEN
C
C RED TARGET KILLED
C TALLY RED KILLS FOR RESOLUTION AFTER THIS SUB-CYCLE
C JUST IN CASE THIS RED TARGET IS ABOUT TO KILL A BLUE
C TARGET
JKILL=1
C
C RE-MODE TO SEARCH
MODEN=1
133 FORMAT ('BLUE',I3,'KILLS RED',I3)
IF (IRSYS(NTG,2).NE.0) WRITE ('',133) K,NTG
IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
  & WRITE ('',1) 'BLUE',K,'WEAPON',NWP,'OUT OF AMMO'
IF (IRSYS(NTG,2).NE.0) WRITE (10,133) K,NTG
IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
  & WRITE (10,1) 'BLUE',K,'WEAPON',NWP,'OUT OF AMMO'
IAMOUT=0
C IF (IRSYS(NTG,2).NE.0) CALL OUTPUT2 (ITIME,2,NTG,0)
ELSE
C
C SOME SORT OF MASKING GOT IN THE WAY OF M-I-L SYSTEM
C OR TERRAIN GOT IN THE WAY FOR F-A-F WEAPON WHILE TARGET
C WAS MOVING
C IF (IRSYS(NTG,2).NE.0) WRITE (10,134) K,NTG
IF (MODEF.EQ.0) GO TO 135
IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
  & WRITE (10,1) 'BLUE',K,'WEAPON',NWP,'OUT OF AMMO'
IAMOUT=0
  & WRITE (10,1) 'BLUE',K,'SEARCHES FOR A BETTER TARGET'
MODEN=1
ENDIF
ELSE
134 FORMAT ('BLUE',I3,'MISSES RED',I3)
IF (IRSYS(NTG,2).NE.0) WRITE ('',134) K,NTG
IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
  & WRITE ('',1) 'BLUE',K,'WEAPON',NWP,'OUT OF AMMO'
IF (IRSYS(NTG,2).NE.0) WRITE (10,134) K,NTG
IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
  & WRITE (10,1) 'BLUE',K,'WEAPON',NWP,'OUT OF AMMO'
IAMOUT=0
IF (MODEF.EQ.0) GO TO 135
RE-MODE BLUE SYSTEM AFTER MISS
THERE ARE TWO OPTIONS -- 1 RE-ENGAGE
-- 2 RE-SEARCH
RE-ENGAGE BASED ON AVAILABLE AMMUNITION, AND THE PK
AND AVAILABLE EM ENERGY
DO NOT RE-ENGAGE IF EM ENERGY IS BELOW FULL PK SHOT

```

```

C RE-SEARCH IF DECISION IS TO NOT RE-ENGAGE
PKIT=PKTB(K,1)
NSHOTS=NSHOTS(K,NWP)
SKILL=LOG(.01)/LOG(1-PKIT)
C EM ENERGY
NSYS=IBSYS(K,1)
IF (NSYS.EQ.NBLUE) THEN
  ENED=ENP(NWP,1)
  IF (ELSYS(K).LT.ENED) THEN
    NSHOTS=0.0
    MODEN=3
  IF (IRSYS(NTG,2).NE.0)
    & WRITE (10,1) 'BLUE',K,'INSUFFICIENT EM ENERGY TO RE-ENGAGE'
  ENDF
ENDIF
IF (NSHOTS.EQ.0) MODEN=1
IF (SKILL.GT.10.0) MODEN=1
IF (SKILL.LE.10.0.AND.SKILL.GT.7.0) THEN
  IF (SKILL.LE.10.0.AND.NSHOTS.NE.0) THEN
    IF (SKILL.GT.7.0.AND.NSHOTS.NE.0) THEN
      IF (NSHOTS.NE.0) THEN
        PSHOOT=NSHOTS/SKILL
        PNO=RN(0)
        IF (SKILL.LE.7) PNO=0.0
        IF (SKILL.LE.4) PNO=0.0
        IF (PNO.GE.PSHOOT.OR.MODEF.EQ.0) THEN
          MODEN=1
        WRITE ('',1) 'BLUE',K,'SEARCHES FOR A BETTER TARGET'
        WRITE (10,1) 'BLUE',K,'SEARCHES FOR A BETTER TARGET'
        ELSE
          IF (IRSYS(NTG,2).NE.0) THEN
            RE-ENGAGE THE TARGET
            MODEN=0
          WRITE ('',1) 'BLUE',K,'RE-ENGAGES RED',NTG
          WRITE (10,1) 'BLUE',K,'RE-ENGAGES RED',NTG
          DETERMINE END TIME OF ENGAGEMENT MODE CYCLE
          BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
          ENGAGEMENT TIMES
          TRNG=PKTB(K,2)
          IST=(NSYS-1)*3*NWP
          NRG=NRWB(NSYS,NWP)
          ISRM=(NSYS-1)*18+(NWP-1)*6*NRG
          ISR1=(NSYS-1)*18+(NWP-1)*6+1
          RMIN=RB(ISR1)
          RMAX=RB(ISRM)
          TMIN=TEMINB(IST)
          TMAX=TEMAB(IST)
          CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
          ELSE
            MODEN=1
          ENDF
        ENDF
      ENDF
135 CONTINUE
ENDIF
C
C END LOOP 10
C
C END OF TARGET KILLED AND SYSTEM RE-MODED
C
C TRMBF(K)=ITIME+IDTIME+STIME
IF (MODEN.EQ.0) MODEN=3
MODEFB(K)=MODEN
IF (MODEF.EQ.0) MODEFB(K)=0
C
C WRITE SHOT OUTPUT INFORMATION
IF (IRSYS(NTG,2).EQ.0) THEN
  JKILL=1
  IF (IFAPS.NE.-1) IFAPS=3
  ENDF
  & CALL OUTPUT1 (ITIME,1,K,NSYS,NWP,ENERGY,TRNG,NTG,IRSYS(NTG,1),
  & IFAPS,JKILL)
C
C ELSE
C
C END OF IT'S TIME TO KILL THE TARGET
C
C CHECK TO SEE IF TARGET WAS KILLED BY ANOTHER WEAPON DURING
C THIS ENGAGEMENT PERIOD. IF IT WAS, CHECK TO SEE IF THIS WEAPON
C CAN REDESIGNATE.
C
NTG=ITGTB(K,1)
IF (IRSYS(NTG,2).EQ.0) THEN
  C THIS TARGET HAS JUST BEEN KILLED, TRY TO RE-DESIGNATE
  C IF THIS SYSTEM JUST RE-DESIGNATED, HOWEVER, MUST WAIT
  C 3 SECOND RE-DESIGNATION TIME PERIOD BEFORE DOING IT AGAIN
  IF (MODEF.EQ.0) THEN
    TCAN=TRDES8(K)
  ELSE
    TCAN=0.0
  ENDF
  NWP=ITGTB(K,2)
  NSYS=IBSYS(K,1)
  MILTYPE=MILB(NSYS,NWP)
  IF (MILTYPE.EQ.1.AND.ITIME.GE.TCAN) THEN
    C THIS WEAPON IS MAN-IN-LOOP, IT CAN RE-DESIGNATE, BUT NEEDS
    C AT LEAST 3 SECONDS FOR GUNNER TO OVERCOME HIS SURPRISE AND
    C QUICKLY ASSESS THE SITUATION, THEN FIND ANOTHER TARGET.
    C FIND TIME TO IMPACT
    TRDES8(K)=ITIME+3.0
    THIT=TRMBF(K)-ITIME
    IWANT=0
    IF (THIT.GT.3.0) THEN
      C THERE IS ENOUGH TIME
    
```

```

THIT=THIT-3.0
TRNG=PKTB(K,2)
C DETERMINE ENGAGEMENT MODE CYCLE TIME TO THIS TARGET
C BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
C ENGAGEMENT TIMES AND TARGET RANGE
MYWP=NWP
IST=(NSYS-1)*3+MYWP
NRG=NRWB(NSYS,MYWP)
SRM=(NSYS-1)*18+(MYWP-1)*8+NRG
ISR1=(NSYS-1)*18+(MYWP-1)*8+1
RMIN=RB(ISR1)
RMAX=RB(ISRM)
TMIN=TEMINB(IST)
TMAX=TEMAXB(IST)
CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
C FIND DISTANCE MISSILE IS FROM TARGET
DMSL=TRNG/STIME*THIT
C MISSILE CAN BE RE-DESIGNATED TO FIND ANOTHER RED
C TARGET WITHIN THIS DISTANCE FROM THE OLD TARGET
C BUT NOT BEYOND SYSTEM MAXIMUM RANGE.
C
C FIND CLOSEST OTHER RED TARGET
C CURRENT TARGET X,Y POSITION
XT=RXA(NTG,1)
YT=RYA(NTG,2)
XB=BXA(K,1)
YB=BYA(K,2)
C SEARCH ALL RED TARGETS FOR CLOSEST
DRMIN=1.0E12
C FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C SO THAT MASKING CAN BE EVALUATED
C
KSIDE=1
MYSIDE=1
IBLUE=K
CALL VMASK (KSIDE,IBLUE)
C
C EVALUATE OBSCURANTS ALSO
C IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION
C OR TRACKING
C FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C RE-EVALUATED
RANGE=TRNG*1000*.1
ITRK1=ITRK8(NSYS,1)
ISROF=(NSYS-1)*3+NWP
IRNG1=IRNG8(ISROF,1)
ICANRD=1
IF (IRNG1.EQ.0) ICANRD=0
IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
& CALL DMASK (XB,YB,RANGE,0.0,0.0)
C
CALL COOKMASK (XB,YB,RANGE,0.0,0.0)
C
C IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C CAN BE RE-EVALUATED
ITRK2=ITRK8(NSYS,2)
ISROF=(NSYS-1)*3+NWP
IRNG2=IRNG8(ISROF,2)
ICANRE=1
IF (IRNG2.EQ.0) ICANRE=0
IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
& CALL ESMASK (XB,YB,RANGE,0.0,0.0)
C
C IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C CAN BE RE-EVALUATED
ITRK3=ITRK8(NSYS,3)
ISROF=(NSYS-1)*3+NWP
IRNG3=IRNG8(ISROF,3)
ICANRS=1
IF (IRNG3.EQ.0) ICANRS=0
IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
& CALL GRMASK (XB,YB,RANGE,0.0,0.0)
C
C IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
C MASKING CAN BE RE-EVALUATED
ITRK4=ITRK8(NSYS,4)
ISROF=(NSYS-1)*3+NWP
IRNG4=IRNG8(ISROF,4)
ICANRA=1
IF (IRNG4.EQ.0) ICANRA=0
IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
& CALL ASMASK (XB,YB,RANGE,0.0,0.0)
C
C DO 11 JRED=1,NR
C IF (IRSYS(JRED,2).EQ.0) GO TO 11
XNT=RXA(JRED,1)
YNT=RYA(JRED,2)
DNT=(XNT-XT)**2+(YNT-YT)**2*.5
DNB=((XNT-XB)**2+(YNT-YB)**2)*.5
RANGE=DNB
C CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C NOT DETECTED
CALL CHECKSIG (1,K,JRED,RANGE,JSKIP)
IF (JSKIP.EQ.0) GO TO 11

IF (DNB.LE.RMAX.AND.DNT.LE.DMSL) THEN
DIVIDE BY A RANDOM FACTOR SO EVERYONE DOES NOT
TAKE THE SAME ALTERNATE TARGET
RRR=RND()
DNT=DNT/RRR
IF (DNT.LT.DRMIN) THEN
C EVALUATE MASKING
C FIND ANGLE TO THIS TARGET
TOP=YNT-YB
BOT=XNT-XB
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP)/ABS(BOT))*180/PI
IF (BOT.LT.0.0) THEN
IF (TOP.LT.0.0) DIR=DIR-180.
IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
IF (TOP.LT.0.0) DIR=360.-DIR
ENDIF
C CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (DIR,AZIN)
C
C AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C MASKING AND OBSCURANTS AND DROP THIS
C TARGET IF UNOBSERVABLE
JSKIP=1
C TERRAIN MASKING
IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XNT,YNT,JSKIP)
IF (JSKIP.EQ.0) GO TO 11
CALL VSEE (MYSIDE,K,JRED,RANGE,AZIN,JSKIP,NB,NR)
C
C BURNING VEHICLE MASKING
CALL COOKSEE (XB,YB,XNT,YNT,AZIN,JSKIP)
IF (JSKIP.EQ.0) GO TO 11
C
C DUST MASKING
C RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
JDUST=1
IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
& CALL DSEE (XB,YB,XNT,YNT,AZIN,JDUST)
C
C IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE OR
C TRACK THROUGH IT, THE TARGET IS MISSED
IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0
C
C IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT BLUE CAN SEE
C THROUGH IT. THE QUESTION REMAINS WHETHER BLUE CAN TRACK THROUGH
C THE DUST.
IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0
C
C IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM
C
C ENGINE SMOKE MASKING
C RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
JENG=1
IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
& CALL ESSEE (XB,YB,XNT,YNT,AZIN,JENG)
IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0
C
C SMOKE GRENADE CURTAIN MASKING
C RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
JSMK=1
IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
& CALL GRSEE (XB,YB,XNT,YNT,AZIN,JSMK)
IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0
C
C ARTILLERY SMOKE CURTAIN MASKING
C RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
C TARGET
JART=1
IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
& CALL ASSEE (XB,YB,XNT,YNT,AZIN,JART)
IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0
C
C END OF OBSCURANT RE-EVALUATION FOR M-I-L WEAPON
IF (JSKIP.EQ.0) GO TO 11
C
C TARGET CAN BE SEEN AND IS CLOSER TO OLD TARGET
DRMIN=DNT
IWANT=JRED
RNG=DNB
ENDIF
ENDIF
11 CONTINUE
ENDIF
IF (IWANT.NE.0) THEN
C THERE IS ANOTHER TARGET TO DESIGNATE
C FIND PROBABILITY OF KILLING THIS NEW RED TARGET
NITSYS=IRSYS(IWANT,1)
CALL FINDPK (KSIDE,NSYS,NITSYS,NWP,NRNG,RNG,PKIT,IWANT)
PKTB(K,1)=PKIT
PKTB(K,2)=RNG
MODEN=8
STIME=THIT+3.0
ITGTB(K,1)=IWANT
TRMFB(K)=ITIME+IDTIME+STIME

```

```

C      MODEFR(K)=MODEN
C      WRITE ('.') 'BLUE 'K' RE-DESIGNATES AT RED 'I,WANT
C      WRITE (10,'.') 'BLUE 'K' RE-DESIGNATES AT RED 'I,WANT
      ELSE
C      NO TARGETS, FINISH OUT CURRENT ENGAGEMENT MODE TIME
C      THEN RE-SEARCH
C      MODEN=1
C      BLUE EXPENDS ONE SHOT
C      NWP=ITGTR(K,2)
C      NTG=ITGTR(K,1)
C      TRNG=PKTR(K,2)
C      NSHOTB(K,NWP)=NSHOTB(K,NWP)-1
C      IAMR(NTG)=IAMR(NTG)-1
C      IF (IAMR(NTG).LT.0) IAMR(NTG)=0
C      FIND SEARCH AREAS FOR RE-MODING INFORMATION
C      NSYS=IRSYS(K,1)
C      ALFT1=IRPRIME(K,1)
C      ARG1=IRPRIME(K,2)
C      ALFT2=IRSECOND(K,1)
C      ARG2=IRSECOND(K,2)
C      IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
C      IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
C      IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
C      EQ.IRSECOND(K,2)) THEN
C      ALFT2=-1.0
C      ARG2=-1.0
C      ENDIF
C      DETERMINE END TIME OF SEARCH MODE CYCLE
C      STIME=((ARG1-ALFT1)+(ARG2-ALFT2)/SRCHB(NSYS)
C      ENDIF
C      TRMFR(K)=ITIME+IDTIME+STIME
C      MODEFR(K)=MODEN
C      WRITE SHOT OUTPUT INFORMATION
C      JKILL=1
C      IFAPS=1
C      CALL OUTPUT1 (ITIME,1,K,NSYS,NWP,0,TRNG,NTG,IRSYS(NTG,1),
C      IFAPS,JKILL)
C      ELSE
C      IF (IAMOUT.EQ.1)
C      & WRITE ('.') 'BLUE 'K' WEAPON 'NWP' OUT OF AMMO'
C      & WRITE (10,'.') 'BLUE 'K' WEAPON 'NWP' OUT OF AMMO'
C      IAMOUT=0
C      ENDIF
C      END OF IT CAN BE RE-DESIGNATED
C      ENDIF
C      END OF THIS TARGET HAS JUST BEEN KILLED
C      ENDIF
C      END LOOP 9
C      END OF REDESIGNATING MODE
C      ENDIF
C      END LOOP 8
C      END OF ENGAGEMENT MODE
150 CONTINUE
C      END LOOP 1
C      END OF BLUE SYSTEM FIRE MODE LOOP
C      NOW DO RED
C      RED
C      DO 1150 K=1,NR
C      LOOP 1
C      CHECK TO SEE IF THIS UNIT IS KILLED AND SKIP
C      WRITE ('.') 'KILL STATUS RED 'K' ALIVE'
C      BUT BEFORE SKIPPING IT, SEE IF IT WAS A FIRE-AND-FORGET
C      WEAPON AND IN THE LAST SECONDS OF ENGAGEMENT MODE OR
C      RE-ENGAGEMENT MODE.
C      MODEFR(K)=MODEFR(K)
C      IF (IRSYS(K,2).EQ.0) THEN
C      NSYS=IRSYS(K,1)
C      IF (MODEFR.EQ.2.OR.MODEFR.EQ.6) THEN
C      NWP=ITGTR(K,2)
C      MILTYPE=MILR(NSYS,NWP)
C      IF (MILTYPE.EQ.0) THEN
C      CALCULATE TIME TO BULLET IMPACT
C      TCHANGE=TRMFR(K)
C      BORE=BORE(NSYS,NWP)/1000.
C      TOF=TOFR(NSYS,NWP)
C      TRNG=PKTR(K,2)
C      TTRNG=TRNG/BORE*TOF+1.0
C      IF (ITIME.GE.(TCHANGE-TTRNG)) THEN
C      THIS WEAPON IS NOW IN SACRIFICE MODE SINCE IT IS DEAD AND
C      HAPPENED TO GET A FIRE-AND-FORGET SHOT OFF FIRST
C      MODEFR=9
C      MODEFR(K)=MODEFR
C      NSHOTR(K,NWP)=NSHOTR(K,NWP)-1
C      CALL OUTPUT2 (ITIME,2,K,0)
C      WRITE ('.') 'RED 'K' MAKES SACRIFICE SHOT'
C      WRITE (10,'.') 'RED 'K' MAKES SACRIFICE SHOT'
C      ELSE
C      MODEFR(K)=0
C      FIRE MODE SET TO 0 -- KILLED WHILE ENGAGING A TARGET
C      ENDIF

```

```

      ENDIF
      ENDIF
      ENDIF
C      IF (IRSYS(K,2).EQ.0.AND.MODEFR.NE.0) GO TO 1150
C      IF (MODEFR.EQ.6) THEN
C      RE-EVALUATE RE-ENGAGEMENT MODE TO SEE IF THIS RED TARGET
C      WAS JUST KILLED
C      IF (IRSYS(ITGTR(K,1),2).EQ.1) THEN
C      MODEFR=6
C      MODEFR(K)=6
C      WRITE (10,'.') 'RED 'K' RE-ENGAGES BLUE 'ITGTR(K,1)
C      ELSE
C      THIS TARGET WAS KILLED LAST GO AROUND SO FIND ANOTHER
C      MODEFR=1
C      MODEFR(K)=1
C      NSYS=IRSYS(K,1)
C      ALFT1=IRPRIME(K,1)
C      ARG1=IRPRIME(K,2)
C      IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
C      ALFT2=IRSECOND(K,1)
C      ARG2=IRSECOND(K,2)
C      IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
C      IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
C      EQ.IRSECOND(K,2)) THEN
C      ALFT2=-1.0
C      ARG2=-1.0
C      ENDIF
C      DETERMINE END TIME OF SEARCH MODE CYCLE
C      STIME=((ARG1-ALFT1)+(ARG2-ALFT2)/SRCHB(NSYS)
C      TRMFR(K)=ITIME+STIME
C      ENDIF
C      ENDIF
C      MODEFR=MODEFR(K)
C      WRITE ('.') 'FIRE MODE RED 'MODEFR
C      INCREASE CYCLE TIME IN THIS MODE
C      TCFMR(K,MODEFR)=TCFMR(K,MODEFR)+DT
C      WRITE ('.') 'CYCLE TIME RED 'TCFMR(K,MODEFR)
C      IF (MODEFR.EQ.1.OR.MODEFR.EQ.4.OR.MODEFR.EQ.5) THEN
C      LOOP 2
C      SEARCH MODE (1) OR BALK MODE (4) OR WAIT MODE (5)
C      BALK MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
C      FOR DUTY CYCLE ACCOUNTABILITY
C      WAIT MODE IS ACTUALLY RE-SEARCHING, BUT IS BROKEN OUT
C      FOR DUTY CYCLE ACCOUNTABILITY
C      CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
C      TCHANGE=TRMFR(K)
C      WRITE ('.') 'FIRE MODE CHANGE RED 'TCHANGE
C      TNEXT=ITIME+IDTIME
C      IF (TCHANGE.LE.TNEXT) THEN
C      LOOP 3
C      IT'S TIME TO FIND A TARGET
C      FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C      SO THAT MASKING CAN BE EVALUATED
C      KSIDE=2
C      IRED=K
C      MYSIDE=2
C      CALL VMASK (KSIDE,IRED)
C      FIND ALL BLUE TARGETS IN PRIMARY AND SECONDARY ZONES
C      OUT TO MAX RANGE FOR EACH ON-BOARD WEAPON
C      NSYS=IRSYS(K,1)
C      NWEAP=NRWR(NSYS)
C      NRNG1=NRWR(NSYS,1)
C      RMAX1=0.0
C      RMAX2=0.0
C      RMAX3=0.0
C      NUM1=0
C      NUM2=0
C      NUM3=0
C      ISR=(1-1)*18+(J-1)*6+K
C      WHERE L=SYSTEM FILE (10 MAX); J=SYSTEM WEAPON (3 MAX);
C      K=WEAPON RANGE (6 MAX); N=TARGET (2*10 MAX)
C      NUM1=(NSYS-1)*18+(1-1)*6+NRNG1
C      RMAX1=RR(NUM1)
C      WRITE ('.') 'RMAX1',RMAX1,RR(22),RR(22)
C      WRITE ('.') 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG1',NRNG1,'NUM1',NUM1
C      WRITE ('.') 'NSHOTR',NSHOTR(K,1)
C      IF (NSHOTR(K,1).LE.0) THEN
C      RMAX1=0.0
C      NUM1=0
C      ENDIF
C      IF (NWEAP.GE.2) THEN
C      NRNG2=NRWR(NSYS,2)
C      NUM2=(NSYS-1)*18+(2-1)*6+NRNG2
C      RMAX2=RR(NUM2)
C      WRITE ('.') 'NSYS',NSYS,'NWEAP',NWEAP,'NRNG2',NRNG2,'NUM2',NUM2
C      WRITE ('.') 'NSHOTR',NSHOTR(K,2)
C      IF (NSHOTR(K,2).LE.0) THEN
C      RMAX2=0.0
C      NUM2=0
C      ENDIF
C      ENDIF
C      IF (NWEAP.EQ.3) THEN
C      NRNG3=NRWR(NSYS,3)
C      NUM3=(NSYS-1)*18+(3-1)*6+NRNG3

```



```

C RMAX3=RR(NUM3)
C WRITE (," NSYS",NSYS,"NWEAP",NWEAP,"NRNG3",NRNG3,"NUM3",NUM3)
C WRITE (," NSHOTR",NSHOTR(K,3))
C IF (NSHOTR(K,3).LE.0) THEN
C   RMAX3=0.0
C   NUM3=0
C   ENDF
C SEARCH ALL ZONES BASED ON ZONE AZIMUTHS
C   IVER1=0
C   IVER2=0
C   ALFT1=IRPRIME(K,1)
C   ARG1=IRPRIME(K,2)
C   ALFT2=IRSECOND(K,1)
C   ARG2=IRSECOND(K,2)
C   IF (ALFT1.GT.ARG1) IVER1=1
C   IF (ALFT2.GT.ARG2) IVER2=1
C   IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
C   IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
C   IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
C     EQ.IRSECOND(K,2)) THEN
C     ALFT2=1.0
C     ARG2=1.0
C     ENDF
C   GET X,Y COORDINATES FOR THIS RED UNIT
C   XB=RXVA(K,1)
C   YB=RYVA(K,2)
C   INITIALIZE PRIORITIES
C   NTGT1=0
C   PRI=0.0
C   MYWP1=0
C   NTGT2=0
C   SEC=0.0
C   MYWP2=0
C   NTGT3=0
C   THRD=0.0
C   MYWP3=0
C   TRNG1=0.0
C   TRNG2=0.0
C   PKWP1=0.0
C   PKWP2=0.0
C   PKWP3=0.0
C IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C EVALUATED
C ITRK1=ITRK1(NSYS,1)
C ISROF=(NSYS-1)*3+1
C IRNG11=1
C IRNG12=1
C IRNG13=1
C IRNG11=IRNGR(ISROF,1)
C IF (NWEAP.GE.2) IRNG12=IRNGR((ISROF+1),1)
C IF (NWEAP.GT.2) IRNG13=IRNGR((ISROF+2),1)
C ICANRD=1
C IF (IRNG11.EQ.0.OR.IRNG12.EQ.0.OR.IRNG13.EQ.0) ICANRD=0
C IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
C   CALL DMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C CALL COOMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C CAN BE EVALUATED
C ITRK2=ITRK2(NSYS,2)
C ISROF=(NSYS-1)*3+1
C IRNG21=1
C IRNG22=1
C IRNG23=1
C IRNG21=IRNGR(ISROF,2)
C IF (NWEAP.GE.2) IRNG22=IRNGR((ISROF+1),2)
C IF (NWEAP.GT.2) IRNG23=IRNGR((ISROF+2),2)
C ICANRE=1
C IF (IRNG21.EQ.0.OR.IRNG22.EQ.0.OR.IRNG23.EQ.0) ICANRE=0
C IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
C   CALL ESMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C CAN BE EVALUATED
C ITRK3=ITRK3(NSYS,3)
C ISROF=(NSYS-1)*3+1
C IRNG31=1
C IRNG32=1
C IRNG33=1
C IRNG31=IRNGR(ISROF,3)
C IF (NWEAP.GE.2) IRNG32=IRNGR((ISROF+1),3)
C IF (NWEAP.GT.2) IRNG33=IRNGR((ISROF+2),3)
C ICANRS=1
C IF (IRNG31.EQ.0.OR.IRNG32.EQ.0.OR.IRNG33.EQ.0) ICANRS=0
C IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
C   CALL GRMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
C MASKING

```

```

C CAN BE EVALUATED
C ITRK4=ITRK4(NSYS,4)
C ISROF=(NSYS-1)*3+1
C IRNG41=1
C IRNG42=1
C IRNG43=1
C IRNG41=IRNGR(ISROF,4)
C IF (NWEAP.GE.2) IRNG42=IRNGR((ISROF+1),4)
C IF (NWEAP.GT.2) IRNG43=IRNGR((ISROF+2),4)
C ICANRA=1
C IF (IRNG41.EQ.0.OR.IRNG42.EQ.0.OR.IRNG43.EQ.0) ICANRA=0
C IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
C   CALL ASMASK (XB,YB,RMAX1,RMAX2,RMAX3)
C DO 1149 J=1,NB
C   LOOP 4
C   CHECK TO SEE IF THIS BLUE UNIT IS KILLED
C   IF (ISYS(J,2).EQ.0.OR.ISYS(J,1).GT.10) GO TO 1149
C   FIND THIS BLUE UNIT'S X,Y COORDINATES
C   XR=BXVA(J,1)
C   YR=BYVA(J,2)
C   FIND RANGE TO THIS TARGET
C   RANGE=((XR-XB)**2+(YR-YB)**2)**.5
C   WRITE (," RANGE TO BLUE ",J,RANGE)
C CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C NOT DETECTED
C CALL CHECKSIG (2,K,J,RANGE,JSKIP)
C IF (JSKIP.EQ.0) GO TO 1149
C IF THIS TARGET IS OUTSIDE THE RANGE OF ALL WEAPONS SKIP IT
C WRITE (," RANGE,RANGE,RMAX1,RMAX1,RMAX2,RMAX2,
C   RMAX3,RMAX3
C   JSKIP=0
C   IF ((RANGE*1000.).LE.RMAX1) JSKIP=1
C   IF ((RANGE*1000.).LE.RMAX2) JSKIP=1
C   IF ((RANGE*1000.).LE.RMAX3) JSKIP=1
C   WRITE (," RANGE SKIP ",JSKIP," 0=YES")
C   IF (JSKIP.EQ.0) GO TO 1149
C   TARGET IS WITHIN RANGE OF ONE OF THESE WEAPONS
C   IF THIS TARGET IS OUTSIDE THE FIRE ORDERS RANGE WINDOW SKIP IT
C   IF (RANGE.LT.WRNGR(K,1).OR.RANGE.GT.WRNGR(K,2)) JSKIP=0
C   IF (JSKIP.EQ.0) GO TO 1149
C   FIND ANGLE TO THIS TARGET
C   TOP=YR-YB
C   BOT=XR-XB
C   WRITE (," BOT",BOT)
C   IF (BOT.EQ.0.0) BOT=BOT+.001
C   DIR=ATAN2(TOP/ABS(BOT))*180./PI
C   IF (BOT.LT.0.0) THEN
C     IF (TOP.LT.0.0) DIR=DIR+180.
C     IF (TOP.GT.0.0) DIR=180.-DIR
C   ELSE
C     IF (TOP.LT.0.0) DIR=360.-DIR
C   ENDF
C   CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C   CALL FAZ (DIR,AZIN)
C AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C MASKING AND OBSCURANTS AND DROP THIS
C TARGET FROM ZONE IF UNOBSERVABLE (SMOKE NOT DONE YET)
C TERRAIN MASKING
C IF (ITER.EQ.1) CALL TERRMASK (XR,YR,XB,YB,JSKIP)
C IF (JSKIP.EQ.0) GO TO 1149
C VEHICLE MASKING
C CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C BURNING VEHICLE MASKING
C CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)
C IF (JSKIP.EQ.0) GO TO 1149
C DUST MASKING
C IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
C JDUST=1
C IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
C   CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C   IF THERE IS DUST IN THE WAY AND RED CANNOT SEE THROUGH IT
C   SKIP THIS TARGET
C   IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 1149
C   THERE MAY BE DUST IN THE WAY, BUT RED CAN SEE THROUGH IT.
C   THE QUESTION REMAINS WHETHER RED CAN RANGE OR TRACK THROUGH
C   THE DUST. IF JDUST=0 THERE IS DUST IN THE WAY.
C ENGINE SMOKE MASKING
C IF THIS RED SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C JENG=1
C IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
C   CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
C   IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 1149
C SMOKE GRENADE CURTAIN MASKING
C IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
C JSMK=1

```

```

IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
  & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
  IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 1149
C
ARTILLERY SMOKE MASKING
C
IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
ACQUISITION
C
TRACKING OR RANGING
C
EVALUATE ALL ARTILLERY SMOKE PUFFS WITHIN RANGE OF THIS TARGET
  JART=1
  IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
    & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
    IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 1149
C
  INZONE=3
  IF (AZINLE.ALFT1.AND.AZINLE.ARG1) INZONE=1
    IF (OVER1.EQ.1.AND.AZINLE.(ARG1-360.0)) INZONE=1
  C
  TARGET IS IN PRIMARY ZONE
  IF (AZINLE.ALFT2.AND.AZINLE.ARG2) INZONE=2
    IF (OVER2.EQ.1.AND.AZINLE.(ARG2-360.0)) INZONE=2
  C
  TARGET IS IN SECONDARY ZONE
  AT THIS POINT EVALUATE TERRAIN MASKING AND OBSCURANTS AND DROP
  TARGET FROM ZONE IF UNOBSERVABLE (NOT DONE YET)
C
  WRITE ('. ') 'BLUE '.J. 'INZONE '.INZONE
  IF (INZONE.GE.1) THEN
    LOOP 5
    GIVE THIS TARGET A PRIORITY NUMBER
    AND IN THE END SELECT ONLY THE HIGHEST PRIORITY TARGET
    TARGET PRIORITY IS A COMPOSITE OF THE PROBABILITY OF THIS
    TARGET KILLING YOU TIMES YOUR PROBABILITY OF KILLING IT.
    NITSYS=ISSYS(J,1)
    CYCLE THROUGH ALL WEAPONS ON YOUR SYSTEM FOR THE ONE WITH THE
    HIGHEST PK AGAINST THIS TARGET AND EVALUATE WHETHER YOU WILL
    SHOOT IT BASED ON NUMBER OF AVAILABLE ROUNDS OF AMMUNITION
    AND THE PK AGAINST THIS TARGET
    PKIT1=0.0
    PKIT2=0.0
    PKIT3=0.0
    KSIDE=2
    PKITM=0.0
    IWEAP=0
    DO 1141 NUM=1,3
      LOOP 6
      IF (NUM.EQ.1.AND.NUM1.EQ.0) GO TO 1141
      IF (NUM.EQ.2.AND.NUM2.EQ.0) GO TO 1141
      IF (NUM.EQ.3.AND.NUM3.EQ.0) GO TO 1141
      IF (NUM.EQ.1) THEN
        CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG1,RANGE,PKIT1,J)
        IF (ISPOR(K).GT.0.OR.ISPOB(J).GT.0) THEN
          MILTYPE=MILR(NSYS,NUM)
          IF (MILTYPE.EQ.0) THEN
            BORE=BORER(NSYS,NUM)
            TOF=TOFR(NSYS,NUM)
            CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
          ENDIF
        ENDIF
        IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
          IF (IRNG1.EQ.0.OR.IRNG2.EQ.0) THEN
            THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
            EVALUATE M-I-L OR F-A-F
            MILTYPE=MILR(NSYS,NUM)
            IF (MILTYPE.EQ.0) THEN
              THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
              BORE=BORER(NSYS,NUM)
              TOF=TOFR(NSYS,NUM)
              CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT1)
            ELSE
              SKIP THIS WEAPON SINCE CANT TRACK THE TARGET
              PKIT1=0.0
            ENDIF
          ENDIF
        ENDIF
        NSHOTS=NSHOTR(K,NUM)
        WRITE ('. ') 'LOG BOT1'.LOG(1.-PKIT1),PKIT1,PKIT1
        IF (PKIT1.NE.0.0) THEN
          SKILL=LOG(.01)/LOG(1.-PKIT1)
        ELSE
          SKILL=9999.
        ENDIF
        IF (NSHOTS.EQ.0) PKIT1=0.0
        IF (SKILL.GT.16.0) PKIT1=0.0
        IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
          IF (SKILL.GT.7.0) THEN
            PSHOOT=NSHOTS/SKILL
            PNO=RND()
            IF (PNO.GE.PSHOOT) PKIT1=0.0
          ENDIF
        IF (PKIT1.GT.PKITM) THEN
          PKITM=PKIT1
          IWEAP=1
        ENDIF
      ENDIF
      IF (NUM.EQ.2) THEN
        CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG2,RANGE,PKIT2,J)
        IF (ISPOR(K).GT.0.OR.ISPOB(J).GT.0) THEN
          MILTYPE=MILR(NSYS,NUM)
          IF (MILTYPE.EQ.0) THEN
            BORE=BORER(NSYS,NUM)
            TOF=TOFR(NSYS,NUM)
            CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
          ENDIF
        ENDIF
        IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
          IF (IRNG12.EQ.0.OR.IRNG22.EQ.0) THEN
            THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
            EVALUATE M-I-L OR F-A-F
            MILTYPE=MILR(NSYS,NUM)
            IF (MILTYPE.EQ.0) THEN
              THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
              BORE=BORER(NSYS,NUM)
              TOF=TOFR(NSYS,NUM)
              CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
            ELSE
              SKIP THIS WEAPON SINCE CANT TRACK THE TARGET
              PKIT2=0.0
            ENDIF
          ENDIF
        ENDIF
        NSHOTS=NSHOTR(K,NUM)
        WRITE ('. ') 'LOG BOT2'.LOG(1.-PKIT2),PKIT2,PKIT2
        IF (PKIT2.NE.0.0) THEN
          SKILL=LOG(.01)/LOG(1.-PKIT2)
        ELSE
          SKILL=9999.
        ENDIF
        IF (NSHOTS.EQ.0) PKIT2=0.0
        IF (SKILL.GT.16.0) PKIT2=0.0
        IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
          IF (SKILL.GT.7.0) THEN
            PSHOOT=NSHOTS/SKILL
            PNO=RND()
            IF (PNO.GE.PSHOOT) PKIT2=0.0
          ENDIF
        IF (PKIT2.GT.PKITM) THEN
          PKITM=PKIT2
          IWEAP=2
        ENDIF
      ENDIF
      IF (NUM.EQ.3) THEN
        CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG3,RANGE,PKIT3,J)
        IF (ISPOR(K).GT.0.OR.ISPOB(J).GT.0) THEN
          MILTYPE=MILR(NSYS,NUM)
          IF (MILTYPE.EQ.0) THEN
            BORE=BORER(NSYS,NUM)
            TOF=TOFR(NSYS,NUM)
            CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
          ENDIF
        ENDIF
        IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
          IF (IRNG13.EQ.0.OR.IRNG23.EQ.0) THEN
            THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
            EVALUATE M-I-L OR F-A-F
            MILTYPE=MILR(NSYS,NUM)
            IF (MILTYPE.EQ.0) THEN
              THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
              BORE=BORER(NSYS,NUM)
              TOF=TOFR(NSYS,NUM)
              CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
            ELSE
              SKIP THIS WEAPON SINCE CANT TRACK THE TARGET
              PKIT3=0.0
            ENDIF
          ENDIF
        ENDIF
        NSHOTS=NSHOTR(K,NUM)
        WRITE ('. ') 'LOG BOT3'.LOG(1.-PKIT3),PKIT3,PKIT3
        IF (PKIT3.NE.0.0) THEN
          SKILL=LOG(.01)/LOG(1.-PKIT3)
        ELSE
          SKILL=9999.
        ENDIF
        IF (NSHOTS.EQ.0) PKIT3=0.0
        IF (SKILL.GT.16.0) PKIT3=0.0
        IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
          IF (SKILL.GT.7.0) THEN
            PSHOOT=NSHOTS/SKILL
            PNO=RND()
            IF (PNO.GE.PSHOOT) PKIT3=0.0
          ENDIF
        IF (PKIT3.GT.PKITM) THEN
          PKITM=PKIT3
          IWEAP=3
        ENDIF
      ENDIF
    1141 CONTINUE
  END LOOP 6
C
END OF RED WEAPON SELECTION LOOP
WRITE ('. ') 'RED WEAPON SELECTION'.IWEAP,PK,PKITM
C
IF TARGET IS AT A QUESTIONABLE EFFECTIVE RANGE SKIP IT
IF (PKITM.LE.0.0) GO TO 1143
IF (IWEAP.EQ.0.0) GO TO 1143
C
FIND PROBABILITY OF THIS TARGET KILLING YOU AND APPLY PRIORITY
C
FACTOR
C
KPFAC=KPRIR(K,NITSYS)
PKYOU1=0.0
PKYOU2=0.0

```

```

ENDIF
ENDIF
IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
  IF (IRNG12.EQ.0.OR.IRNG22.EQ.0) THEN
    THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
    EVALUATE M-I-L OR F-A-F
    MILTYPE=MILR(NSYS,NUM)
    IF (MILTYPE.EQ.0) THEN
      THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
      BORE=BORER(NSYS,NUM)
      TOF=TOFR(NSYS,NUM)
      CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT2)
    ELSE
      SKIP THIS WEAPON SINCE CANT TRACK THE TARGET
      PKIT2=0.0
    ENDIF
  ENDIF
ENDIF
NSHOTS=NSHOTR(K,NUM)
WRITE ('. ') 'LOG BOT2'.LOG(1.-PKIT2),PKIT2,PKIT2
IF (PKIT2.NE.0.0) THEN
  SKILL=LOG(.01)/LOG(1.-PKIT2)
ELSE
  SKILL=9999.
ENDIF
IF (NSHOTS.EQ.0) PKIT2=0.0
IF (SKILL.GT.16.0) PKIT2=0.0
IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
  IF (SKILL.GT.7.0) THEN
    PSHOOT=NSHOTS/SKILL
    PNO=RND()
    IF (PNO.GE.PSHOOT) PKIT2=0.0
  ENDIF
IF (PKIT2.GT.PKITM) THEN
  PKITM=PKIT2
  IWEAP=2
ENDIF
ENDIF
IF (NUM.EQ.3) THEN
  CALL FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG3,RANGE,PKIT3,J)
  IF (ISPOR(K).GT.0.OR.ISPOB(J).GT.0) THEN
    MILTYPE=MILR(NSYS,NUM)
    IF (MILTYPE.EQ.0) THEN
      BORE=BORER(NSYS,NUM)
      TOF=TOFR(NSYS,NUM)
      CALL PKMOVE (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
    ENDIF
  ENDIF
  IF (JDUST.EQ.0.OR.JENG.EQ.0) THEN
    IF (IRNG13.EQ.0.OR.IRNG23.EQ.0) THEN
      THERE IS A DUST OR ENGINE SMOKE RANGING PROBLEM WITH THIS WEAPON
      EVALUATE M-I-L OR F-A-F
      MILTYPE=MILR(NSYS,NUM)
      IF (MILTYPE.EQ.0) THEN
        THIS WEAPON IS F-A-F, DEGRADE PK BASED ON NO RANGE KNOWLEDGE
        BORE=BORER(NSYS,NUM)
        TOF=TOFR(NSYS,NUM)
        CALL PKLESS (KSIDE,K,J,RANGE,BORE,TOF,PKIT3)
      ELSE
        SKIP THIS WEAPON SINCE CANT TRACK THE TARGET
        PKIT3=0.0
      ENDIF
    ENDIF
  ENDIF
  NSHOTS=NSHOTR(K,NUM)
  WRITE ('. ') 'LOG BOT3'.LOG(1.-PKIT3),PKIT3,PKIT3
  IF (PKIT3.NE.0.0) THEN
    SKILL=LOG(.01)/LOG(1.-PKIT3)
  ELSE
    SKILL=9999.
  ENDIF
  IF (NSHOTS.EQ.0) PKIT3=0.0
  IF (SKILL.GT.16.0) PKIT3=0.0
  IF (SKILL.LE.16.0.AND.SKILL.GT.7.0) THEN
    IF (SKILL.GT.7.0) THEN
      PSHOOT=NSHOTS/SKILL
      PNO=RND()
      IF (PNO.GE.PSHOOT) PKIT3=0.0
    ENDIF
  IF (PKIT3.GT.PKITM) THEN
    PKITM=PKIT3
    IWEAP=3
  ENDIF
ENDIF
ENDIF
1141 CONTINUE
C
END LOOP 6
C
END OF RED WEAPON SELECTION LOOP
WRITE ('. ') 'RED WEAPON SELECTION'.IWEAP,PK,PKITM
C
IF TARGET IS AT A QUESTIONABLE EFFECTIVE RANGE SKIP IT
IF (PKITM.LE.0.0) GO TO 1143
IF (IWEAP.EQ.0.0) GO TO 1143
C
FIND PROBABILITY OF THIS TARGET KILLING YOU AND APPLY PRIORITY
C
FACTOR
C
KPFAC=KPRIR(K,NITSYS)
PKYOU1=0.0
PKYOU2=0.0

```



```

MODEN=2
MODEFR(K)=MODEN
DETERMINE END TIME OF ENGAGEMENT MODE CYCLE
BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
ENGAGEMENT TIMES
IST=(NSYS-1)*3*MYWP
NRG=NRWR(NSYS,MYWP)
SRM=(NSYS-1)*18*(MYWP-1)*8*NRG
SR1=(NSYS-1)*18*(MYWP-1)*8*1
RMIN=RR(SR1)
RMAX=RR(SRM)
TMIN=TEMNR(ST)
TMAX=TEMAR(IST)
CALL TSHOOT(RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
IF TARGET TO BE ENGAGED IS IN ZONE 3 (360 DEGREES),
MUST ADD THIS EXTRA SEARCH TIME TO THE ENGAGEMENT TIME
STIME3=360.0/SRCHR(NSYS)*W3
TRMFR(K)=ITIME-IDTIME+STIME+STIME3
1132 FORMAT 'RED ',I3,' ENGAGES BLUE ',I3,' WEAPON ',I3,
& ' RANGE ',F6.0)
C WRITE ('',1132) K,ITGTR(K,1),MYWP,TRNG*1000.
C WRITE (10,1132) K,ITGTR(K,1),MYWP,TRNG*1000.
ENDIF
ENDIF
END LOOP 7
END OF TARGET SEARCH EVALUATION RESULTS
ENDIF
END LOOP 3
END OF ITS TIME TO SELECT A TARGET
ENDIF
END LOOP 2
END OF SEARCH MODE
IF (MODEF.EQ.2.OR.MODEF.EQ.6.OR.MODEF.EQ.9.OR.
& MODEF.EQ.8) THEN
C LOOP 8
C ENGAGEMENT MODE
C CHECK FOR MODE CHANGE AT END OF SUB-CYCLE
TCHANGE=TRMFR(K)
TNEXT=ITIME-IDTIME
IF (TCHANGE.LE.TNEXT) THEN
C LOOP 9
C ITS TIME TO KILL THE TARGET
C RED EXPENDS ONE SHOT
IAMOUT=0
JKILL=0
NWP=ITGTR(K,2)
NTG=ITGTR(K,1)
TRNG=PKTR(K,2)
NSHOTR(K,NWP)=NSHOTR(K,NWP)-1
IF (NSHOTR(K,NWP).EQ.0) IAMOUT=1
IAMB(NTG)=IAMB(NTG)-1
IF (IAMB(NTG).LE.0) IAMB(NTG)=0
C FIND SEARCH AREAS FOR RE-MODING INFORMATION
NSYS=IRSYS(K,1)
ALFT1=IRPRIME(K,1)
ARGT1=IRPRIME(K,2)
ALFT2=IRSECOND(K,1)
ARGT2=IRSECOND(K,2)
IF (ALFT1.GT.ARG1) ARG1=ARGT1+360.
IF (ALFT2.GT.ARG2) ARG2=ARGT2+360.
IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
& EQ.IRSECOND(K,2)) THEN
C ALFT2=-1.0
C ARG2=-1.0
C ENDIF
C DETERMINE END TIME OF SEARCH MODE CYCLE
STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2)/SRCHR(NSYS)
C RE-EVALUATE MASKING AND OBSCURANT
C PROB=RND()
C PKIT=PKTR(K,1)
C CHECK BLUE TARGET FOR ACTIVE PROTECTION
CALL CHECKAPS (2,NSYS,NTG,NWP,PKIT,IFAPS,TRNG)
C XR=RXYA(K,1)
C YR=RYXA(K,2)
C XB=BXYA(NTG,1)
C YB=BYXA(NTG,2)
C DRAW LINE FROM SHOOTER TO TARGET
C CALL PLINE (XR,YR,XB,YB,2)
C IF (PROB.LE.PKIT) THEN
C LOOP 10
C KILL TARGET AND RE-MODE
C RE-EVALUATE MASKING AND OBSCURANTS
C TERRAIN MASKING
C JSKIP=1
C IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
C IF THIS IS A MAN IN THE LOOP (MIL) WEAPON, THE TARGET
C MUST STILL BE ABLE TO BE SEEN AND TRACKED IN ORDER TO
C BE HIT. EVALUATE CURRENT DUST AND ENGINE SMOKE SITUATION.
C EVALUATE M-L OR F-A-F
C MILTYPE=MILR(NSYS,NWP)
C IF (MILTYPE.EQ.1) THEN
C THIS WEAPON IS M-L, RE-EVALUATE OBSCURANTS FOR MASKING
C IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION

```

```

C OR TRACKING
C FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C RE-EVALUATED
C RANGE=TRNG*1000.
C ITRK1=ITRK1(NSYS,1)
C ISROF=(NSYS-1)*3*NWP
C IRNG1=IRNGR(ISROF,1)
C ICANRD=1
C IF (IRNG1.EQ.0) ICANRD=0
C IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
& CALL DMASK (XR,YR,RANGE,0.0,0.0)
C CALL COOKMASK (XR,YR,RANGE,0.0,0.0)
C IF THIS RED SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C CAN BE RE-EVALUATED
C ITRK2=ITRK2(NSYS,2)
C ISROF=(NSYS-1)*3*NWP
C IRNG2=IRNGR(ISROF,2)
C ICANRE=1
C IF (IRNG2.EQ.0) ICANRE=0
C IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
& CALL ESMASK (XR,YR,RANGE,0.0,0.0)
C IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C CAN BE RE-EVALUATED
C ITRK3=ITRK3(NSYS,3)
C ISROF=(NSYS-1)*3*NWP
C IRNG3=IRNGR(ISROF,3)
C ICANRE=1
C IF (IRNG3.EQ.0) ICANRS=0
C IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
& CALL GRMASK (XR,YR,RANGE,0.0,0.0)
C IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C ACQUISITION,
C TRACKING OR RANGING,
C FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
C MASKING
C CAN BE RE-EVALUATED
C ITRK4=ITRK4(NSYS,4)
C ISROF=(NSYS-1)*3*NWP
C IRNG4=IRNGR(ISROF,4)
C ICANRA=1
C IF (IRNG4.EQ.0) ICANRA=0
C IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
& CALL ASMASK (XR,YR,RANGE,0.0,0.0)
C DUST MASKING
C RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
C FIND ANGLE TO THIS TARGET
C TOP=YB-YR
C BOT=XB-XR
C IF (BOT.EQ.0.0) BOT=BOT+.001
C DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
C IF (BOT.LT.0.0) THEN
C IF (TOP.LT.0.0) DIR=DIR+180.
C IF (TOP.GT.0.0) DIR=180.-DIR
C ELSE
C IF (TOP.LT.0.0) DIR=360.-DIR
C ENDIF
C CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C CALL FAZ (DIR,AZIN)
C JDUST=1
C IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
& CALL DSEE (XR,YR,XB,YB,AZIN,JDUST)
C BURNING VEHICLE MASKING
C CALL COOKSEE (XR,YR,XB,YB,AZIN,JSKIP)
C IF THERE IS DUST IN THE WAY AND RED CANNOT SEE OR
C TRACK THROUGH IT, THE TARGET IS MISSED
C IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0
C IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT RED CAN SEE
C THROUGH IT. THE QUESTION REMAINS WHETHER RED CAN TRACK THROUGH
C THE DUST.
C IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0
C IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM
C ENGINE SMOKE MASKING
C RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C JENG=1
C IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
& CALL ESSEE (XR,YR,XB,YB,AZIN,JENG)
C IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
C IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0
C SMOKE GRENADE CURTAIN MASKING
C RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
C JSMK=1
C IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
& CALL GRSEE (XR,YR,XB,YB,AZIN,JSMK)
C IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0

```

```

IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0
C ARTILLERY SMOKE CURTAIN MASKING
C RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
C TARGET
    JART=1
    IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
    & CALL ASSEE (XR,YR,XB,YB,AZIN,JART)
    IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
    IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0
C ENDF
C END OF OBSCURANT RE-EVALUATION FOR M-L WEAPON
    IF (JSKIP.EQ.1) THEN
C C C BLUE TARGET KILLED
C C C TALLY BLUE KILLS FOR RESOLUTION AFTER THIS SUB-CYCLE
C C C JUST IN CASE THIS BLUE TARGET IS ABOUT TO KILL A RED
C C C TARGET
        KILLR(NTG)=1
        JKILL=1
        RE-MODE TO SEARCH
        MODEN=1
1133 FORMAT ('RED ',J3,' KILLS BLUE ',J3)
        IF (IBSYS(NTG,2).NE.0) WRITE (' ',1133) K,NTG
        IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
        & WRITE (' ',J3,' WEAPON ',NWP,' OUT OF AMMO'
        IF (IBSYS(NTG,2).NE.0) WRITE (' ',1133) K,NTG
        IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
        & WRITE (' ',J3,' WEAPON ',NWP,' OUT OF AMMO'
        IAMOUT=0
        C IF (IBSYS(NTG,2).NE.0) CALL OUTPUT2 (ITIME,1,NTG,0)
        ELSE
C C C SOME SORT OF MASKING GOT IN THE WAY OF M-L SYSTEM
C C C OR TERRAIN GOT IN THE WAY FOR F-A-F WEAPON WHILE TARGET
C C C WAS MOVING
        IF (IBSYS(NTG,2).NE.0) WRITE (' ',1134) K,NTG
        IF (MODEF.EQ.0) GO TO 1135
        IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
        & WRITE (' ',J3,' WEAPON ',NWP,' OUT OF AMMO'
        IAMOUT=0
        WRITE (' ',J3,' SEARCHES FOR A BETTER TARGET'
        MODEN=1
        ENDF
        ELSE
1134 FORMAT ('RED ',J3,' MISSES BLUE ',J3)
        IF (IBSYS(NTG,2).NE.0) WRITE (' ',1134) K,NTG
        IF (IBSYS(NTG,2).NE.0) WRITE (' ',1134) K,NTG
        IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
        & WRITE (' ',J3,' WEAPON ',NWP,' OUT OF AMMO'
        IF (IAMOUT.EQ.1.AND.MODEF.NE.0)
        & WRITE (' ',J3,' WEAPON ',NWP,' OUT OF AMMO'
        IAMOUT=0
        IF (MODEF.EQ.0) GO TO 1135
        RE-MODE RED SYSTEM AFTER MISS
        THERE ARE TWO OPTIONS -- 1 RE-ENGAGE
        -- 2 RE-SEARCH
        RE-ENGAGE BASED ON AVAILABLE AMMUNITION, AND THE PK
        RE-SEARCH IF DECISION IS TO NOT RE-ENGAGE
        PKIT=PKTR(K,1)
        NSHOTS=NSHOTR(K,NWP)
        SKILL=LOG(01)/LOG(1-PKIT)
        IF (NSHOTS.EQ.0) MODEN=1
        IF (SKILL.GT.18.0) MODEN=1
        IF (SKILL.LE.18.0.AND.SKILL.GT.7.0) THEN
        IF (SKILL.LE.18.0.AND.NSHOTS.NE.0) THEN
        IF (SKILL.GT.7.0.AND.NSHOTS.NE.0) THEN
        IF (NSHOTS.NE.0) THEN
            PSHOOT=NSHOTS/SKILL
            PNO=RND0
            IF (SKILL.LE.7) PNO=0.0
            IF (PNO.GE.PSHOOT.OR.MODEF.EQ.0) THEN
                MODEN=1
            C WRITE (' ',J3,' SEARCHES FOR A BETTER TARGET'
            C WRITE (' ',J3,' SEARCHES FOR A BETTER TARGET'
            ELSE
                IF (IBSYS(NTG,2).NE.0) THEN
                    RE-ENGAGE THE TARGET
                    MODEN=0
                C WRITE (' ',J3,' RE-ENGAGES BLUE ',NTG
                C WRITE (' ',J3,' RE-ENGAGES BLUE ',NTG
                C DETERMINE END TIME OF ENGAGEMENT MODE CYCLE
                C BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
                C ENGAGEMENT TIMES
                TRNG=PKTR(K,2)
                IST=(NSYS-1)*3-NWP
                NRG=NRWR(NSYS,NWP)
                ISRM=(NSYS-1)*18-(NWP-1)*8-NRG
                ISR1=(NSYS-1)*18-(NWP-1)*8-1
                RMIN=RR(ISR1)
                RMAX=RR(ISRM)
                TMIN=TEMINR(IST)
                TMAX=TEMAXR(IST)
                CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
                ELSE

```

```

        MODEN=1
        ENDF
        ENDF
1135 CONTINUE
        ENDF
        END LOOP 10
        C END OF TARGET KILLED AND SYSTEM RE-MODED
        TRMFR(K)=ITIME-IDTIME+STIME
        MODEFR(K)=MODEN
        IF (MODEF.EQ.0) MODEFR(K)=0
        C WRITE SHOT OUTPUT INFORMATION
        IF (IBSYS(NTG,2).EQ.0) THEN
            JKILL=1
            IF (IFAPS.NE.-1) IFAPS=3
        ENDF
        & CALL OUTPUT1 (ITIME,2,K,NSYS,NWP,0,TRNG,NTG,IBSYS(NTG,1),
        IFAPS,JKILL)
        C ELSE
        C END OF IT'S TIME TO KILL THE TARGET
        C C C CHECK TO SEE IF TARGET WAS KILLED BY ANOTHER WEAPON DURING
        C C C THIS ENGAGEMENT PERIOD. IF IT WAS, CHECK TO SEE IF THIS WEAPON
        C C C CAN REDESIGNATE.
        C C C NTG=ITGTR(K,1)
        C C C IF (IBSYS(NTG,2).EQ.0) THEN
        C C C THIS WEAPON HAS JUST BEEN KILLED, TRY TO RE-DESIGNATE
        C C C IF THIS SYSTEM JUST RE-DESIGNATED, HOWEVER, MUST WAIT
        C C C 3 SECOND RE-DESIGNATION TIME PERIOD BEFORE DOING IT AGAIN
        C C C IF (MODEF.EQ.0) THEN
            TCAN=TRDES(K)
        ELSE
            TCAN=0.0
        ENDF
        NWP=ITGTR(K,2)
        NSYS=IRSYS(K,1)
        MILTYPE=MILR(NSYS,NWP)
        IF (MILTYPE.EQ.1.AND.ITIME.GE.TCAN) THEN
            THIS WEAPON IS MAN-IN-LOOP, IT CAN RE-DESIGNATE, BUT NEEDS
            AT LEAST 3 SECONDS FOR GUNNER TO OVERCOME HIS SURPRISE AND
            QUICKLY ASSESS THE SITUATION, THEN FIND ANOTHER TARGET.
            FIND TIME TO IMPACT
            TRDES(K)=ITIME+3.0
            THIT=TRMFR(K)-ITIME
            WANT=0
            IF (THIT.GT.3.0) THEN
                THERE IS ENOUGH TIME
                THIT=THIT-3.0
                TRNG=PKTR(K,2)
                DETERMINE ENGAGEMENT MODE CYCLE TIME TO THIS TARGET
                BASED ON INTERPOLATING BETWEEN MIN RANGE AND MAX RANGE
                ENGAGEMENT TIMES AND TARGET RANGE
                MYWP=NWP
                IST=(NSYS-1)*3-MYWP
                NRG=NRWR(NSYS,MYWP)
                ISRM=(NSYS-1)*18-(MYWP-1)*8-NRG
                ISR1=(NSYS-1)*18-(MYWP-1)*8-1
                RMIN=RR(ISR1)
                RMAX=RR(ISRM)
                TMIN=TEMINR(IST)
                TMAX=TEMAXR(IST)
                CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
                FIND DISTANCE MISSILE IS FROM TARGET
                DMSL=TRNG/STIME-THIT
                MISSILE CAN BE RE-DESIGNATED TO FIND ANOTHER RED
                TARGET WITHIN THIS DISTANCE FROM THE OLD TARGET
                BUT NOT BEYOND SYSTEM MAXIMUM RANGE.
                FIND CLOSEST OTHER RED TARGET
                CURRENT TARGET X,Y POSITION
                XT=BXIA(NTG,1)
                YT=BXIA(NTG,2)
                XB=RXIA(K,1)
                YB=RXIA(K,2)
                SEARCH ALL RED TARGETS FOR CLOSEST
                DRMIN=1.0E12
                FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
                SO THAT MASKING CAN BE EVALUATED
                C KSIDE=2
                C MYSIDE=2
                C IRED=K
                C CALL VMASK (KSIDE,IRED)
                C EVALUATE OBSCURANTS ALSO
                C IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION
                C OR TRACKING
                C FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
                C RE-EVALUATED
                C RANGE=TRNG*1000.*1.1
                C ITRK1=ITRK1(NSYS,1)
                C ISROF=(NSYS-1)*3-NWP
                C IRNG1=IRNGR(ISROF,1)
                C ICANRD=1
                C IF (IRNG1.EQ.0) ICANRD=0
                C IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
                & CALL DMASK (XB,YB,RANGE,0.0,0.0)

```

```

C      CALL COOKMASK (XB,YB,RANGE,0.0,0.0)
C
C      IF THIS RED SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C      CAN BE RE-EVALUATED
C      ITRK2=ITRK1(NSYS,2)
C      ISROF=(NSYS-1)*3+NWP
C      IRNG2=IRNG1(ISROF,2)
C      ICANRE=1
C      IF (IRNG2.EQ.0) ICANRE=0
C      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
C      & CALL ESMASK (XB,YB,RANGE,0.0,0.0)
C
C      IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C      CAN BE RE-EVALUATED
C      ITRK3=ITRK1(NSYS,3)
C      ISROF=(NSYS-1)*3+NWP
C      IRNG3=IRNG1(ISROF,3)
C      ICANRS=1
C      IF (IRNG3.EQ.0) ICANRS=0
C      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
C      & CALL GRMASK (XB,YB,RANGE,0.0,0.0)
C
C      IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C      ACQUISITION,
C      TRACKING OR RANGING,
C      FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C      CAN BE RE-EVALUATED
C      ITRK4=ITRK1(NSYS,4)
C      ISROF=(NSYS-1)*3+NWP
C      IRNG4=IRNG1(ISROF,4)
C      ICANRA=1
C      IF (IRNG4.EQ.0) ICANRA=0
C      IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
C      & CALL ASMASK (XB,YB,RANGE,0.0,0.0)
C
C      DO 111 JBLUE=1,NR
C      IF (IBSYS(JBLUE,2).EQ.0) GO TO 111
C      XNT=BXIA(JBLUE,1)
C      YNT=BYIA(JBLUE,2)
C      DNT=((XNT-XT)*2+(YNT-YT)*2)**.5
C      DNB=((XNT-XB)*2+(YNT-YB)*2)**.5
C      RANGE=DNB
C      CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C      NOT DETECTED
C      CALL CHECKSIG (2,K,JBLUE,RANGE,JSKIP)
C      IF (JSKIP.EQ.0) GO TO 111
C      IF (DNB.LE.RMAX.AND.DNT.LE.DMSU) THEN
C      DIVIDE BY A RANDOM FACTOR SO EVERYONE DOES NOT
C      TAKE THE SAME ALTERNATE TARGET
C      RRR=RNDU
C      DNT=DNT/RRR
C      IF (DNT.LT.DRMIN) THEN
C      EVALUATE MASKING
C      FIND ANGLE TO THIS TARGET
C      TOP=YNT-YB
C      BOT=XNT-XB
C      IF (BOT.EQ.0) BOT=BOT+.001
C      DIR=ATAN(ABS(TOP/ABS(BOT)))*180/PI
C      IF (BOT.LT.0) THEN
C      IF (TOP.LT.0) DIR=DIR+180
C      IF (TOP.GT.0) DIR=180-DIR
C      ELSE
C      IF (TOP.LT.0) DIR=360-DIR
C      ENDIF
C      CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C      CALL FAZ (DIR,AZIN)
C
C      AT THIS POINT EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C      MASKING AND OBSCURANTS AND DROP THIS
C      TARGET IF UNOBSERVABLE (TERRAIN / SMOKE NOT DONE YET)
C      JSKIP=1
C      TERRAIN MASKING
C      IF (ITER.EQ.1) CALL TERRMASK (XNT,YNT,XB,YB,JSKIP)
C      IF (JSKIP.EQ.0) GO TO 111
C      CALL VSEE (MYSIDE,K,JBLUE,RANGE,AZIN,JSKIP,NB,NR)
C      CALL COCKSEE (XB,YB,XNT,YNT,AZIN,JSKIP)
C      IF (JSKIP.EQ.0) GO TO 111
C
C      DUST MASKING
C      RE-EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
C      JDUST=1
C      IF (ICANRD.EQ.0.OR.ITRK1.EQ.0)
C      & CALL DSEE (XB,YB,XNT,YNT,AZIN,JDUST)
C
C      IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE OR
C      TRACK THROUGH IT, THE TARGET IS MISSED
C      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) JSKIP=0
C
C      IF JSKIP=1 THERE MAY STILL BE DUST IN THE WAY, BUT BLUE CAN SEE
C      THROUGH IT. THE QUESTION REMAINS WHETHER BLUE CAN TRACK THROUGH
C      THE DUST.
C      IF (JDUST.EQ.0.AND.ICANRD.EQ.0) JSKIP=0
C
C      IF JSKIP=1, THERE IS NO DUST OR TRACKING PROBLEM

```

```

C      ENGINE SMOKE MASKING
C      RE-EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C      JENG=1
C      IF (ICANRE.EQ.0.OR.ITRK2.EQ.0)
C      & CALL ESSEE (XB,YB,XNT,YNT,AZIN,JENG)
C      IF (JENG.EQ.0.AND.ITRK2.EQ.0) JSKIP=0
C      IF (JENG.EQ.0.AND.ICANRE.EQ.0) JSKIP=0
C
C      SMOKE GRENADE CURTAIN MASKING
C      RE-EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
C      JSMK=1
C      IF (ICANRS.EQ.0.OR.ITRK3.EQ.0)
C      & CALL GRSEE (XB,YB,XNT,YNT,AZIN,JSMK)
C      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) JSKIP=0
C      IF (JSMK.EQ.0.AND.ICANRS.EQ.0) JSKIP=0
C
C      ARTILLERY SMOKE CURTAIN MASKING
C      RE-EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS
C      TARGET
C      JART=1
C      IF (ICANRA.EQ.0.OR.ITRK4.EQ.0)
C      & CALL ASSEE (XB,YB,XNT,YNT,AZIN,JART)
C      IF (JART.EQ.0.AND.ITRK4.EQ.0) JSKIP=0
C      IF (JART.EQ.0.AND.ICANRA.EQ.0) JSKIP=0
C
C      END OF OBSCURANT RE-EVALUATION FOR M-I-L WEAPON
C      IF (JSKIP.EQ.0) GO TO 111
C
C      TARGET CAN BE SEEN AND IS CLOSER TO OLD TARGET
C      DRMIN=DNB
C      IWANT=JBLUE
C      RNG=DNB
C      ENDF
C      ENDF
C      111 CONTINUE
C      ENDF
C      IF (IWANT.NE.0) THEN
C      THERE IS ANOTHER TARGET TO DESIGNATE
C      FIND PROBABILITY OF KILLING THIS NEW BLUE TARGET
C      NITSYS=IBSYS(IWANT,1)
C      CALL FINDPK (K,SIDE,NSYS,NITSYS,NWP,NRNG,RNG,PKIT,IWANT)
C      PKTR(K,1)=PKIT
C      PKTR(K,2)=RNG
C      MODEN=8
C      STIME=TIME+.30
C      ITGTR(K,1)=IWANT
C      TRMFR(K)=ITIME+IDTIME+STIME
C      MODEFR(K)=MODEN
C      WRITE ('.') 'RED 'K,' RE-DESIGNATES AT BLUE 'IWANT
C      WRITE (10,') 'RED 'K,' RE-DESIGNATES AT BLUE 'IWANT
C
C      ELSE
C
C      NO TARGETS, FINISH OUT CURRENT ENGAGEMENT MODE TIME
C      THEN RE-SEARCH
C      MODEN=1
C      RED EXPENDS ONE SHOT
C      NWP=ITGTR(K,2)
C      NTG=ITGTR(K,1)
C      TRNG=PKTR(K,2)
C      NSHOTR(K,NWP)=NSHOTR(K,NWP)-1
C      IAMB(NTG)=IAMB(NTG)-1
C      IF (IAMB(NTG).LT.0) IAMB(NTG)=0
C
C      FIND SEARCH AREAS FOR RE-MODING INFORMATION
C      NSYS=IRSYS(K,1)
C      ALFT1=IRPRIME(K,1)
C      ARG1=IRPRIME(K,2)
C      ALFT2=IRSECOND(K,1)
C      ARG2=IRSECOND(K,2)
C      IF (ALFT1.GT.ARG1) ARG1=ARG1+360
C      IF (ALFT2.GT.ARG2) ARG2=ARG2+360
C      IF (IRPRIME(K,1).EQ.IRSECOND(K,1).AND.IRPRIME(K,2).
C      & EQ.IRSECOND(K,2)) THEN
C      ALFT2=1.0
C      ARG2=1.0
C      ENDF
C
C      DETERMINE END TIME OF SEARCH MODE CYCLE
C      STIME=((ARG1-ALFT1)+(ARG2-ALFT2)/SRCHR(NSYS)
C      ENDF
C
C      TRMFR(K)=ITIME+IDTIME+STIME
C      MODEFR(K)=MODEN
C
C      WRITE SHOT OUTPUT INFORMATION
C      JKILL=1
C      IFAPS=1
C      CALL OUTPUT1 (ITIME,2,K,NSYS,NWP,0,TRNG,NTG,IBSYS(NTG,1),
C      & IFAPS,JKILL)
C
C      ELSE
C      IF (IAMOUT.EQ.1)
C      & WRITE ('.') 'RED 'K,' WEAPON 'NWP,' OUT OF AMMO'
C      IF (IAMOUT.EQ.1)
C      & WRITE (10,') 'RED 'K,' WEAPON 'NWP,' OUT OF AMMO'
C      IAMOUT=0
C      ENDF
C      END OF IT CAN BE RE-DESIGNATED
C      ENDF
C      END OF THIS TARGET HAS JUST BEEN KILLED
C

```

```

C      ENDF
C      END OF ITS TIME TO KILL THE TARGET
C      ENDF
C      END OF ENGAGEMENT MODE
1180  CONTINUE
C      END OF RED SYSTEM FIRE MODE LOOP
C      RESOLVE ENGAGEMENTS TO SEE WHO GOT KILLED THIS 60-CYCLE PERIOD
C      DO 900 KILLS=1,NR
C      IF (KILLS(KILLS).EQ.1) THEN
C      IFSYS(KILLS,2)=0
C      CALL OUTPUT2 (ITIME,2,KILLS,0)
C      EVALUATE WHETHER THIS VEHICLE BURNS AND BECOMES AN OBSCURANT
C      CALL COOKS (2,KILLS)
C      KILLS(KILLS)=0
C      ENDF
900  CONTINUE
C      DO 901 KILLS=1,NR
C      IF (KILLS(KILLS).EQ.1) THEN
C      IFSYS(KILLS,2)=0
C      CALL OUTPUT2 (ITIME,1,KILLS,0)
C      CALL COOKS (1,KILLS)
C      KILLS(KILLS)=0
C      ENDF
901  CONTINUE
C      EVALUATE MINEFIELD KILLS
C      IF (NMINES.GT.0) THEN
C      DO 902 MKILLS=1,NR
C      IF (MINEK(MKILLS,2).GT.0) THEN
C      IF (IRSYS(MKILLS,2).NE.0) CALL OUTPUT2 (ITIME,2,MKILLS,0)
C      IFSYS(MKILLS,2)=0
C      CALL WMINE (ITIME,2,MKILLS,MINEK(MKILLS,2))
C      CALL OUTPUT2 (ITIME,2,MKILLS,0)
C      NAZMINE(MINEK(MKILLS,2),3)=1
C      WRITE (10,*) 'RED 'MKILLS' KILLED BY MINEFIELD'.
C      MINEK(MKILLS,2)
C      ENDF
902  CONTINUE
C      DO 903 MKILLS=1,NR
C      IF (MINEK(MKILLS,1).GT.0) THEN
C      IF (IRSYS(MKILLS,1).NE.0) CALL OUTPUT2 (ITIME,1,MKILLS,0)
C      IFSYS(MKILLS,1)=0
C      CALL WMINE (ITIME,1,MKILLS,MINEK(MKILLS,1))
C      CALL OUTPUT2 (ITIME,1,MKILLS,0)
C      NAZMINE(MINEK(MKILLS,1),3)=1
C      WRITE (10,*) 'BLUE 'MKILLS' KILLED BY MINEFIELD'.
C      MINEK(MKILLS,1)
C      ENDF
903  CONTINUE
C      ENDF
C      END OF RESOLVING ENGAGEMENTS
C      IF (ITIME.GE.TPUFF) THEN
C      MOVE OBSCURANTS
C      CALL MPUFFS
C      DELETE DISSIPATED OBSCURANTS
C      CALL DELPUFFS
C      CREATE NEW OBSCURANTS
C      CALL NEWPUFFS
C      ERASE OLD OBSCURANT POSITIONS
C      CALL ERASEPUFFS
C      ERASE OLD SMOKE CURTAIN POSITIONS
C      CALL ERASEMKCURT
C      TPUFF=DTPUFF+ITIME
C      ENDF
C      CHARGE EM SYSTEMS
C      CALL EMCHARGE
C      ITIME=ITIME+IDTIME
C      WRITE (10,*) 'TIME 'ITIME
C      WRITE (10,*) 'TIME 'ITIME
C      ERASE OLD UNIT POSITIONS
C      IF (ICYLE.NE.60) CALL EUNITS
C      CALL WOTHERS (BACEL)
1000 CONTINUE
C      PLOT ONE LAST TIME THE OBSCURANT POSITIONS
C      IF (IDUST.EQ.1.AND.NOUST.GT.0) CALL POUST
C      PLOT ENGINE SMOKE CLOUDS
C      IF (ENG.GT.0) CALL PENG
C      PLOT GRENADE SMOKE CURTAINS
C      IF (NSMK.GT.0) CALL PSMKCURTAINS
C      ARTILLERY SMOKE
C      IF (MART.GT.0) CALL PARTSMK
C      ACCUMULATE MODE CYCLE TIMES
C      DO 1800 L=1,NR
C      TTMMB(L)=TTMMB(L)+TCMMB(L,1)
C      TTMMB(L)=TTMMB(L)+TCMMB(L,2)
C      TTMMB(L)=TTMMB(L)+TCMMB(L,3)

```

```

C      TTMMB(L)=TTMMB(L)+TCFMB(L,1)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,2)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,3)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,4)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,5)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,6)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,7)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,8)
C      TTMMB(L)=TTMMB(L)+TCFMB(L,9)
C      IF (IRSYS(L).EQ.NSLUED) THEN
C      TTEM(L)=TTEM(L)+TEM(L,1)
C      TTEM(L)=TTEM(L)+TEM(L,2)
C      ENDF
C1800 CONTINUE
C      DO 1850 L=1,NR
C      TTMMR(L)=TTMMR(L)+TCMMR(L,1)
C      TTMMR(L)=TTMMR(L)+TCMMR(L,2)
C      TTMMR(L)=TTMMR(L)+TCMMR(L,3)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,1)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,2)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,3)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,4)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,5)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,6)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,7)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,8)
C      TTMMR(L)=TTMMR(L)+TCFMR(L,9)
C1850 CONTINUE
C      WRITE BLUE SYSTEM #1 (EM TANK) MODE TIMES FOR THIS
C      SIXTY CYCLE PERIOD
C      CALL WMODES
C      READ (10,*)
C      CALL PLOT (0.0,999)
C      RETURN
C      ENDF
C      SUBROUTINE TIME
C      COMMON/BATTLE/ITIME,IDTIME,ITGTS(200,2),ITGTR(200,2),
C      & PKTB(200,2),PKTR(200,2),KILLS(200),KILLR(200),IAMB(200),
C      & IAMB(200)
C      WRITE (10,*)
C      WRITE (10,*) 'CURRENT TIME STEP 'IDTIME
C      WRITE (10,*) 'ENTER NEW TIME STEP (INTEGER)'
C      WRITE (10,*)
C      READ (10,*) ERR=1) DTIME
C      IDTIME=INT(DTIME)
C      WRITE (10,*) 'NEW TIME STEP 'IDTIME
C      WRITE (10,*) 'RETURN TO CONTINUE'
C      READ (10,*)
C      RETURN
C      ENDF
C      SUBROUTINE FIREORS
C      FIREORS.FOR
C      ENTER FIRE CONTROL ORDERS FOR UNITS
C      COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
C      COMMON/TOURS/X(1100),Y(1100),X2(1100),Y2(1100),IC(1100),
C      & NC(15),NCOL(15),JCON
C      COMMON/ZOOM/BLX,BLY,XLEN,ZFACT,N,XMAX,YMAX,YUNIT
C      COMMON/FORCE/BIHER(200),PHIER(200),BUNIT(200),RUNIT(200),
C      & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,L2RF,L3RF,
C      COMMON/MOVE/SXYA(200,2),RXYA(200,2),IRSA(200,2),IRSA(200,2),
C      & L1BM,L2M,L3M,L1RM
C      COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
C      & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,2),PKR(180,2),
C      & NRNDB(30),NRNDR(30),TEMAB(30),TEMAR(30),TEMNB(30),
C      & TEMNR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
C      & MILB(10,3),MILR(10,3)
C      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
C      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
C      COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
C      & BXYP(200,2),RXYP(200,2),ISPOB(200),ISPOR(200)
C      COMMON/FIRE/IRPRIME(200,2),IRSECON(200,2),IRPRIME(200,2),
C      & IRSECON(200,2),WRNGB(200,2),WRNGR(200,2)
C      COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
C      & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
C      & MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
C      & TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMMR(200),
C      & TRMFR(200),TRDES(200),TRDES(200)
C      COMMON/BATTLE/ITIME,IDTIME,ITGTS(200,2),ITGTR(200,2),
C      & PKTB(200,2),PKTR(200,2),KILLS(200),KILLR(200),IAMB(200),
C      & IAMB(200)
C      COMMON/TYPES/NBAT,NCOMP,ICOMP(10),NPLAT(10),IPLAT(10,10),
C      & NSQD(10,10),JPLAT,JSQD
C      CHARACTER*9 BFNAME,RFNAME
C      CHARACTER*30 LINE
C      CHARACTER*16 BIHER,PHIER,BIHER
C      CHARACTER*24 BUNIT,RUNIT
C      CHARACTER*30 L1BF,L2F,L3F,L1RF,L2RF,L3RF,L1BM,L2M,L3M,L1RM
C      CHARACTER*1 C2
C      CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C      FIRE CONTROL ORDERS
C      WRITE (10,*) 3

```

```

C WRITE (," (PS(ISR),ISR-1,38)
C WRITE (," (RR(ISR),ISR-1,38)
C
C KSIDE=0
10 WRITE (," FIRE CONTROL ORDERS
WRITE (," enter BLUE or RED, or QUIT
WRITE (,"
READ (," ERR-10) C2
IF (C2(1:1).EQ.'B') KSIDE=1
IF (C2(1:1).EQ.'R') KSIDE=2
IF (C2(1:1).EQ.'Q') GO TO 1000
IF (KSIDE.EQ.0) GO TO 10
C
8 WRITE (,"
WRITE (," CURRENT STATUS
FORMAT (1X,A50)
IF (C2(1:1).EQ.'B') WRITE (," 1) L1BM
IF (C2(1:1).EQ.'R') WRITE (," 1) L1RM
WRITE (," UNIT HIERARCHY IDENTITY PRIMARY
&ZONE SECONDARY ZONE
WRITE (,"
&right left right left
C IF (C2(1:1).EQ.'B') THEN
C BLUE
KK=0
DO 130 K=1,NB
IF (IBSYS(K,1).GT.10) GO TO 130
121 FORMAT (1X,3,1X,A18,A24,A(18,1X))
IF (IBSYS(K,2).EQ.1) THEN
WRITE (," 121) K,HIER(K),BUNIT(K),BPRIME(K,1),BPRIME(K,2),
& BSECOND(K,1),BSECOND(K,2)
KK=KK+1
ENDIF
IF (KK.EQ.20) THEN
READ (,"
KK=0
ENDIF
130 CONTINUE
C
C ELSE
C RED
KK=0
DO 140 K=1,NR
IF (IRSYS(K,1).GT.10) GO TO 140
& IRSECOND(K,1),IRSECOND(K,2)
KK=KK+1
ENDIF
IF (KK.EQ.20) THEN
READ (,"
KK=0
ENDIF
140 CONTINUE
ENDIF
141 WRITE (," ENTER COMMAND UNIT (0-RELISTS -1-QUIT)
WRITE (,"
READ (," ERR-141) ICOMMAND
IF (ICOMMAND.EQ.0) GO TO 5
IF (ICOMMAND.EQ.-1) GO TO 10
C
142 WRITE (," ENTER NUMBER RANGE OF SUBORDINATES FOR
WRITE (," SUB-UNIT ALLOCATION (#START #END) (0 0 FOR NONE)
WRITE (,"
READ (," ERR-142) NSUB1,NSUB2
NSUB=1
IF (NSUB1.LE.0) THEN
NSUB1=0
NSUB2=0
NSUB=0
ENDIF
IF (NSUB2.LE.0) THEN
NSUB1=0
NSUB2=0
NSUB=0
ENDIF
IF (NSUB.EQ.0) GO TO 145
C
143 WRITE (," SUB-UNIT ALLOCATIONS
WRITE (," 1-ALL UNITS SEQUENTIAL INCLUDING COMMANDERS
WRITE (," 2-ALL SUBORDINATES HAVE SAME ENTIRE SECTOR AS
WRITE (," 3-ALL SUBORDINATES HAVE SAME ENTIRE SECTOR AS
WRITE (," THE COMMANDER (EVERYONE SHOTS IN SAME PLACE)
WRITE (," 4-UNIT COMMANDERS AND PLATOON LEADERS HAVE
WRITE (," THEIR WHOLE SUBORDINATE UNIT SECTOR
WRITE (,"
READ (," ERR-143) ISUB
IF (ISUB.NE.1.AND.ISUB.NE.2) GO TO 143
C
145 WRITE (,"
WRITE (," ENTER PRIMARY ZONE AZIMUTHS (LEFT TO RIGHT)
WRITE (," FOR COMMAND UNIT AND SECONDARY ZONE AZIMUTHS
WRITE (," ENTER 0 0 IF NO SECONDARY ZONE (Integers)
WRITE (,"
READ (," ERR-145) IPL,IPR,ISL,ISR
ISEC=1
IF (IPL.LT.0) IPL=IPL+360
IF (IPR.LT.0) IPR=IPR+360
IF (ISL.LT.0) ISL=ISL+360
IF (ISR.LT.0) ISR=ISR+360

```

```

IF (ISL.EQ.0.AND.ISR.EQ.0) ISEC=0
PL=IPL
PR=IPR
SL=ISL
SR=ISR
IF (ISEC.EQ.0) GO TO 156
C
150 WRITE (,"
WRITE (," SECONDARY SECTOR ALLOCATION
IF (IPL.NE.ISL.AND.IPR.NE.ISR) THEN
WRITE (," 1-UNIQUE COMMAND UNIT ANGLES, SAME SUB-UNIT ALLOC.
ISTYPE=1
ENDIF
IF (IPL.EQ.ISL.AND.IPR.EQ.ISR) THEN
151 WRITE (," ENTER
WRITE (," 2-EVERYONE SHIFT ONE SUB-UNIT ZONE LEFT
WRITE (," 3-EVERYONE SHIFT ONE SUB-UNIT ZONE RIGHT
WRITE (,"
READ (," ERR-151) ISTYLE
IF (ISTYPE.EQ.1) ISTYLE=0
ENDIF
WRITE (,"
IF (ISTYPE.LE.0.OR.ISTYLE.GE.4) GO TO 150
C
155 CONTINUE
C
C ESTABLISH RANGE WINDOWS
200 WRITE (,"
WRITE (," RANGE WINDOWS
WRITE (," 1 - UNRESTRICTED
WRITE (," 2 - ESTABLISH MIN and MAX ENGAGEMENT RANGES
WRITE (,"
READ (," ERR-200) IRNG
IF (IRNG.LT.1.OR.IRNG.GT.2) GO TO 200
IF (IRNG.EQ.2) THEN
WRITE (,"
WRITE (," ENTER MIN RANGE AND MAX RANGE meters
READ (," ERR-200) XMN,XXM
WRITE (," XMN,XXM
XXM=XXM/1000.
XXM=XXM/1000.
ELSE
XMN=0.0
XXM=1.0E8
ENDIF
C
C BLUE
IF (KSIDE.EQ.1) THEN
WRNGB(ICOMMAND,1)=XMN
WRNGB(ICOMMAND,2)=XXM
BPRIME(ICOMMAND,1)=IPL
BPRIME(ICOMMAND,2)=IPR
IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
MODEFB(ICOMMAND)=7
ELSE
IF (MODEFB(ICOMMAND).EQ.7.OR.MODEFB(ICOMMAND).EQ.1) THEN
C DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
ALFT1=IPL
ARGT1=IPR
IF (ALFT1.GT.ARG1) ARG1=ARGT1+360.
ALFT2=ISL
ARGT2=ISR
IF (ALFT2.GT.ARG2) ARG2=ARGT2+360.
NSYS=IBSYS(ICOMMAND,1)
STIME=(ARGT1-ALFT1)+(ARGT2-ALFT2)/SRCHB(NSYS)
TSHOOT=0.0
IF (MODEFB(ICOMMAND).EQ.7) TSHOOT=STIME*RND0
IF (MODEFB(ICOMMAND).EQ.1) TSHOOT=TRMFB(ICOMMAND)-ITIME
TRMFB(ICOMMAND)=ITIME+STIME+TSHOOT
MODEFB(ICOMMAND)=1
ENDIF
ENDIF
IF (ISEC.EQ.0) THEN
IBSECOND(ICOMMAND,1)=0
IBSECOND(ICOMMAND,2)=0
ELSE
IBSECOND(ICOMMAND,1)=ISL
IBSECOND(ICOMMAND,2)=ISR
ENDIF
IF (NSUB.NE.0) THEN
C IF (ISUB.EQ.1) THEN
C ALLOCATE SEQUENTIALLY TO ALL SUBORDINATES
PANGLE=PR-PL
SANGLE=SR-SL
MSUB=0
DO 157 L=NSUB1,NSUB2
IF (IBSYS(L,2).EQ.1.AND.IBSYS(L,1).LE.10) THEN
WRNGB(L)=XMN
WRNGB(L)=XXM
MSUB=MSUB+1
IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
MODEFB(L)=7
ELSE
IF (MODEFB(L).EQ.7.OR.MODEFB(L).EQ.1) THEN
C DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
C SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
ALFT1=IPL
ARGT1=IPR
IF (ALFT1.GT.ARG1) ARG1=ARGT1+360.
ALFT2=ISL

```



```

ARGT2=ISR
IF (ALFT2.GT.ARG2) ARGT2=ARGT2+360.
NSYS=ISYS(1)
STIME=((ARGT1-ALFT1)+(ARGT2-ALFT2))/SRCHB(NSYS)
TSHOOT=0.0
IF (MODEFR(1).EQ.7) TSHOOT=STIME*RND()
IF (MODEFR(1).EQ.1) TSHOOT=TRMFR(1)-ITIME
TRMFR(1)=ITIME+STIME+TSHOOT
MODEFR(1)=1
ENDIF
ENDIF
187 CONTINUE
DPANGLE=PANGLE/MSUB
DSANGLE=SANGLE/MSUB
C IC=0
DO 170 I=NSUB1,NSUB2
  IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 170
  IC=IC+1
  IPRIME(1,1)=INT(PL+(IC-1)*DPANGLE)
  IPRIME(1,2)=INT(PL+(IC-1)*DPANGLE)
  IF (ISEC.EQ.0) THEN
    ISECOND(1,1)=0
    ISECOND(1,2)=0
  ELSE
    ISECOND(1,1)=INT(SL+(IC-1)*DSANGLE)
    ISECOND(1,2)=INT(SL+(IC-1)*DSANGLE)
  ENDIF
170 CONTINUE
  IF (ISTYPE.EQ.2) THEN
    DO 171 I=NSUB1,NSUB2
      IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 171
      IF (I.NE.NSUB1) THEN
        ISECOND(1,1)=IPRIME(1,1)
        ISECOND(1,2)=IPRIME(1,2)
      ELSE
        ISECOND(1,1)=INT(SL-DSANGLE)
        ISECOND(1,2)=INT(SL-DSANGLE)
      ENDIF
      IF (ISECOND(1,1).LT.0) ISECOND(1,1)=ISECOND(1,1)+360
      ISECOND(1,2)=ISL
    ENDIF
  LAST=I
  CONTINUE
  ENDIF
  IF (ISTYPE.EQ.3) THEN
    IDEL=NSUB2-NSUB1+1
    DO 172 J=1,IDEL
      I=NSUB2-J+1
      IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 172
      IF (I.NE.NSUB2) THEN
        ISECOND(1,1)=IPRIME(1,1)
        ISECOND(1,2)=IPRIME(1,2)
      ELSE
        ISECOND(1,1)=ISR
        ISECOND(1,2)=INT(SL+DSANGLE)
      ENDIF
      IF (IRSECOND(1,2).GT.360) IRSECOND(1,2)=IRSECOND(1,2)-360
    ENDIF
  LAST=I
  CONTINUE
  ENDIF
172 CONTINUE
  ENDIF
C
C IF (SUB.EQ.2) THEN
  EVERY ONE GETS THE SAME ZONES
  DO 175 I=NSUB1,NSUB2
    WRNGB(1,1)=XMN
    WRNGB(1,2)=XMX
    IPRIME(1,1)=IPRIME(1,1)
    IPRIME(1,2)=IPRIME(1,2)
    IF (ISYS(1,2).EQ.1.AND.ISYS(1,1).LE.10) THEN
      IF (IPLEQ.0.AND.IPREQ.0) THEN
        MODEFR(1)=7
      ELSE
        IF (MODEFR(1).EQ.7.OR.MODEFR(1).EQ.1) THEN
          DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
          SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
          ALFT1=IPR
          ARG1=IPR
          IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
          ALFT2=ISL
          ARG2=ISR
          IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
          NSYS=ISYS(1,1)
          STIME=((ARG1-ALFT1)+(ARG2-ALFT2))/SRCHB(NSYS)
          TSHOOT=0.0
          IF (MODEFR(1).EQ.7) TSHOOT=STIME*RND()
          IF (MODEFR(1).EQ.1) TSHOOT=TRMFR(1)-ITIME
          TRMFR(1)=ITIME+STIME+TSHOOT
          MODEFR(1)=1
        ENDIF
        ENDIF
        167 CONTINUE
        DPANGLE=PANGLE/MSUB
        DSANGLE=SANGLE/MSUB
        C IC=0
        DO 180 I=NSUB1,NSUB2
          IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 180
          IC=IC+1
          IPRIME(1,1)=INT(PL+(IC-1)*DPANGLE)
          IPRIME(1,2)=INT(PL+(IC-1)*DPANGLE)
          IF (ISEC.EQ.0) THEN
            ISECOND(1,1)=0
            ISECOND(1,2)=0
          ELSE
            ISECOND(1,1)=INT(SL+(IC-1)*DSANGLE)
            ISECOND(1,2)=INT(SL+(IC-1)*DSANGLE)
          ENDIF
          180 CONTINUE
            IF (ISTYPE.EQ.2) THEN
              DO 181 I=NSUB1,NSUB2
                IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 181
                IF (I.NE.NSUB1) THEN
                  ISECOND(1,1)=IPRIME(1,1)
                  ISECOND(1,2)=IPRIME(1,2)
                ELSE
                  ISECOND(1,1)=INT(SL-DSANGLE)
                  ISECOND(1,2)=INT(SL-DSANGLE)
                ENDIF
              LAST=I
              CONTINUE
              ENDIF
            181 CONTINUE
              ENDIF
          ENDIF
        ENDIF
        IF (ISEC.EQ.0) THEN
          ISECOND(1,1)=0
          ISECOND(1,2)=0
        ELSE
          ISECOND(1,1)=ISECOND(1,1)
          ISECOND(1,2)=ISECOND(1,2)
        ENDIF
        175 CONTINUE
        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

```

ENDIF
ENDIF
C
C RED
IF (KSIDE.EQ.2) THEN
  WRNGB(1,1)=XMN
  WRNGB(1,2)=XMX
  IPRIME(1,1)=IPR
  IPRIME(1,2)=IPR
  IF (IPLEQ.0.AND.IPREQ.0) THEN
    MODEFR(1)=7
  ELSE
    IF (MODEFR(1).EQ.7.OR.MODEFR(1).EQ.1) THEN
      DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
      SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
      ALFT1=IPR
      ARG1=IPR
      IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
      ALFT2=ISL
      ARG2=ISR
      IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
      NSYS=ISYS(1,1)
      STIME=((ARG1-ALFT1)+(ARG2-ALFT2))/SRCHB(NSYS)
      TSHOOT=0.0
      IF (MODEFR(1).EQ.7) TSHOOT=STIME*RND()
      IF (MODEFR(1).EQ.1) TSHOOT=TRMFR(1)-ITIME
      TRMFR(1)=ITIME+STIME+TSHOOT
      MODEFR(1)=1
    ENDIF
    ENDIF
    IF (ISEC.EQ.0) THEN
      IRSECOND(1,1)=0
      IRSECOND(1,2)=0
    ELSE
      IRSECOND(1,1)=ISL
      IRSECOND(1,2)=ISR
    ENDIF
    IF (NSUB.NE.0) THEN
      IF (SUB.EQ.1) THEN
        C
        ALLOCATE UNIFORMLY TO ALL SUBORDINATES
        PANGLE=PR-PL
        SANGLE=SR-SL
        MSUB=0
        DO 167 I=NSUB1,NSUB2
          IF (ISYS(1,2).EQ.1.AND.ISYS(1,1).LE.10) THEN
            WRNGB(1,1)=XMN
            WRNGB(1,2)=XMX
            MSUB=MSUB+1
            IF (IPLEQ.0.AND.IPREQ.0) THEN
              MODEFR(1)=7
            ELSE
              IF (MODEFR(1).EQ.7.OR.MODEFR(1).EQ.1) THEN
                DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
                SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
                ALFT1=IPR
                ARG1=IPR
                IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
                ALFT2=ISL
                ARG2=ISR
                IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
                NSYS=ISYS(1,1)
                STIME=((ARG1-ALFT1)+(ARG2-ALFT2))/SRCHB(NSYS)
                TSHOOT=0.0
                IF (MODEFR(1).EQ.7) TSHOOT=STIME*RND()
                IF (MODEFR(1).EQ.1) TSHOOT=TRMFR(1)-ITIME
                TRMFR(1)=ITIME+STIME+TSHOOT
                MODEFR(1)=1
              ENDIF
              ENDIF
              167 CONTINUE
                DPANGLE=PANGLE/MSUB
                DSANGLE=SANGLE/MSUB
                C IC=0
                DO 180 I=NSUB1,NSUB2
                  IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 180
                  IC=IC+1
                  IPRIME(1,1)=INT(PL+(IC-1)*DPANGLE)
                  IPRIME(1,2)=INT(PL+(IC-1)*DPANGLE)
                  IF (ISEC.EQ.0) THEN
                    ISECOND(1,1)=0
                    ISECOND(1,2)=0
                  ELSE
                    ISECOND(1,1)=INT(SL+(IC-1)*DSANGLE)
                    ISECOND(1,2)=INT(SL+(IC-1)*DSANGLE)
                  ENDIF
                  180 CONTINUE
                    IF (ISTYPE.EQ.2) THEN
                      DO 181 I=NSUB1,NSUB2
                        IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 181
                        IF (I.NE.NSUB1) THEN
                          ISECOND(1,1)=IPRIME(1,1)
                          ISECOND(1,2)=IPRIME(1,2)
                        ELSE
                          ISECOND(1,1)=INT(SL-DSANGLE)
                          ISECOND(1,2)=INT(SL-DSANGLE)
                        ENDIF
                      LAST=I
                      CONTINUE
                      ENDIF
                    181 CONTINUE
                      ENDIF
                  ENDIF
                ENDIF
                IF (ISEC.EQ.0) THEN
                  ISECOND(1,1)=0
                  ISECOND(1,2)=0
                ELSE
                  ISECOND(1,1)=ISECOND(1,1)
                  ISECOND(1,2)=ISECOND(1,2)
                ENDIF
                175 CONTINUE
                ENDIF
              ENDIF
            ENDIF
          ENDIF
        ENDIF
        IF (ISEC.EQ.0) THEN
          ISECOND(1,1)=0
          ISECOND(1,2)=0
        ELSE
          ISECOND(1,1)=ISECOND(1,1)
          ISECOND(1,2)=ISECOND(1,2)
        ENDIF
        175 CONTINUE
        ENDIF
      ENDIF
    ENDIF
  ENDIF
  IF (SUB.EQ.2) THEN
    EVERY ONE GETS THE SAME ZONES
    DO 175 I=NSUB1,NSUB2
      WRNGB(1,1)=XMN
      WRNGB(1,2)=XMX
      IPRIME(1,1)=IPRIME(1,1)
      IPRIME(1,2)=IPRIME(1,2)
      IF (ISYS(1,2).EQ.1.AND.ISYS(1,1).LE.10) THEN
        IF (IPLEQ.0.AND.IPREQ.0) THEN
          MODEFR(1)=7
        ELSE
          IF (MODEFR(1).EQ.7.OR.MODEFR(1).EQ.1) THEN
            DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
            SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
            ALFT1=IPR
            ARG1=IPR
            IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
            ALFT2=ISL
            ARG2=ISR
            IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
            NSYS=ISYS(1,1)
            STIME=((ARG1-ALFT1)+(ARG2-ALFT2))/SRCHB(NSYS)
            TSHOOT=0.0
            IF (MODEFR(1).EQ.7) TSHOOT=STIME*RND()
            IF (MODEFR(1).EQ.1) TSHOOT=TRMFR(1)-ITIME
            TRMFR(1)=ITIME+STIME+TSHOOT
            MODEFR(1)=1
          ENDIF
          ENDIF
          167 CONTINUE
            DPANGLE=PANGLE/MSUB
            DSANGLE=SANGLE/MSUB
            C IC=0
            DO 180 I=NSUB1,NSUB2
              IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 180
              IC=IC+1
              IPRIME(1,1)=INT(PL+(IC-1)*DPANGLE)
              IPRIME(1,2)=INT(PL+(IC-1)*DPANGLE)
              IF (ISEC.EQ.0) THEN
                ISECOND(1,1)=0
                ISECOND(1,2)=0
              ELSE
                ISECOND(1,1)=INT(SL+(IC-1)*DSANGLE)
                ISECOND(1,2)=INT(SL+(IC-1)*DSANGLE)
              ENDIF
              180 CONTINUE
                IF (ISTYPE.EQ.2) THEN
                  DO 181 I=NSUB1,NSUB2
                    IF (ISYS(1,2).NE.1.OR.ISYS(1,1).GT.10) GO TO 181
                    IF (I.NE.NSUB1) THEN
                      ISECOND(1,1)=IPRIME(1,1)
                      ISECOND(1,2)=IPRIME(1,2)
                    ELSE
                      ISECOND(1,1)=INT(SL-DSANGLE)
                      ISECOND(1,2)=INT(SL-DSANGLE)
                    ENDIF
                  LAST=I
                  CONTINUE
                  ENDIF
                181 CONTINUE
                  ENDIF
              ENDIF
            ENDIF
            IF (ISEC.EQ.0) THEN
              ISECOND(1,1)=0
              ISECOND(1,2)=0
            ELSE
              ISECOND(1,1)=ISECOND(1,1)
              ISECOND(1,2)=ISECOND(1,2)
            ENDIF
            175 CONTINUE
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF

```

```

IF (ISTYPE.EQ.3) THEN
  IDEL=NSUB2-NSUB1+1
  DO 182 J=1,IDEL
    L=NSUB2-(J-1)
    IF (IRSYS(L2).NE.1.OR.IRSYS(L1).GT.10) GO TO 182
    IF (L.NE.NSUB2) THEN
      IRSECOND(L1)=RPRIME(LAST.1)
      IRSECOND(L2)=RPRIME(LAST.2)
    ELSE
      IRSECOND(L1)=ISR
      IRSECOND(L2)=INT(SL+DSANGLE)
    ENDIF
  182  LAST=L
  CONTINUE
ENDIF
C
C
IF (ISUB.EQ.2) THEN
  EVERY ONE GETS THE SAME ZONES
  DO 186 L=NSUB1,NSUB2
    WRNGR(L1)=XMIN
    WRNGR(L2)=XMX
    IRPRIME(L1)=RPRIME(ICOMMAND.1)
    IRPRIME(L2)=RPRIME(ICOMMAND.2)
    IF (IRSYS(L2).EQ.1.AND.IRSYS(L1).LE.10) THEN
      IF (IPL.EQ.0.AND.IPR.EQ.0) THEN
        MODEFR(I)=7
      ELSE
        IF (MODEFR(I).EQ.7.OR.MODEFR(I).EQ.1) THEN
          DETERMINE END TIME OF SEARCH MODE CYCLE IF ALREADY IN
          SEARCH MODE OR ENTERING SEARCH MODE FROM SURVEILLANCE MODE
          ALFT1=IPL
          ARG1=IPR
          IF (ALFT1.GT.ARG1) ARG1=ARG1+360.
          ALFT2=ISR
          ARG2=ISR
          IF (ALFT2.GT.ARG2) ARG2=ARG2+360.
          NSYS=IRSYS(L1)
          STIME=((ARG1-ALFT1)+(ARG2-ALFT2)/SRCHR(NSYS))
          TSHOOT=0.0
          IF (MODEFR(I).EQ.7) TSHOOT=STIME*RNDO
          IF (MODEFR(I).EQ.1) TSHOOT=TRMFR(I)-TIME
          TRMFR(I)=TIME+STIME+TSHOOT
          MODEFR(I)=1
        ENDIF
      ENDIF
    ENDIF
    IF (ISEC.EQ.0) THEN
      IRSECOND(L1)=0
      IRSECOND(L2)=0
    ELSE
      IRSECOND(L1)=IRSECOND(ICOMMAND.1)
      IRSECOND(L2)=IRSECOND(ICOMMAND.2)
    ENDIF
  186  CONTINUE
ENDIF
C
C
GO TO 141
1000 CONTINUE
RETURN
END
SUBROUTINE COMMAND
10  WRITE ('.')
  WRITE ('.') 1 - MOVEMENT ORDERS
  WRITE ('.') 2 - DIRECT-FIRE ORDERS
  WRITE ('.') 3 - VEHICLE OBSCURANTS ORDERS
  WRITE ('.') 4 - DIRECT-FIRE TARGET PRIORITIES
  WRITE ('.') 5 - INDIRECT-FIRE ORDERS
  WRITE ('.') 6 - QUIT
  WRITE ('.')
  READ ('.',ERR=10) IWANT
  IF (IWANT.EQ.1) CALL MOVEORS
  IF (IWANT.EQ.2) CALL FIREORS
  IF (IWANT.EQ.3) CALL OBSCURANTS
  IF (IWANT.EQ.4) CALL PRIORITIES
  IF (IWANT.EQ.5) CALL IDFIREORS
  IF (IWANT.EQ.6) RETURN
  GO TO 10
END
C
SUBROUTINE FINDPK (KSIDE,NSYS,NITSYS,NUM,NRNG,RT,PKIT,NUMT)
  INTERPOLATES PK TABLES TO GIVE PK AT DESIGNATED RANGE
  C
  COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
  & NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
  & NRNG(30),NRND(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
  & TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
  & MLB(10,3),MLR(10,3)
  COMMON/DEFILADE/DEFB(200),IDFR(200)
  DIMENSION PK(6),R(6)
  CHARACTER*8 BFNAME,RFNAME
  C
  C
  WRITE ('.') 4
  C
  WRITE ('.') (RB(ISR),ISR=1,36)

```

```

C
WRITE ('.') (RR(ISR),ISR=1,36)
C
C
KSIDE 1 - BLUE 2 - RED
NSYS ATTACKING SYSTEM NUMBER
NITSYS TARGET SYSTEM NUMBER
NUM ATTACKING WEAPON NUMBER
NRNG NUMBER OF RANGES IN RANGE TABLE
RT TARGET RANGE
PKIT WEAPON PK ON THIS TARGET AT THIS RANGE
NUMT TARGET NUMBER
C
PKIT=0.0
IF (NSYS.GT.10) RETURN
IF (NITSYS.GT.10) THEN
  PKIT=1.0
  RETURN
ENDIF
RANGE=RT*.999*1000.
DO 10 NR=1,6
  PK(NR)=0.0
  R(NR)=0.0
10  CONTINUE
C
DO 100 NR=1,NRNG
  IF (KSIDE.EQ.1) THEN
    N=(NSYS-1)*18+(NUM-1)*6+NR
    R(NR)=R(N)
  C
  DETERMINE IF TARGET IS EXPOSED OR IN DEFILADE
  JDEF=IDFR(NUMT)
  IF (JDEF.EQ.0) THEN
    TARGET IS EXPOSED
    NIT=(NITSYS-1)*2+1
    PK(NR)=PKB(N,NIT)
  ELSE
    C
    TARGET IN DEFILADE
    NIT=(NITSYS-1)*2+2
    PK(NR)=PKB(N,NIT)
    ENDF
  ELSE
    N=(NSYS-1)*18+(NUM-1)*6+NR
    R(NR)=R(N)
  C
  DETERMINE IF TARGET IS EXPOSED OR IN DEFILADE
  JDEF=IDFR(NUMT)
  IF (JDEF.EQ.0) THEN
    TARGET IS EXPOSED
    NIT=(NITSYS-1)*2+1
    PK(NR)=PKR(N,NIT)
  ELSE
    C
    TARGET IN DEFILADE
    NIT=(NITSYS-1)*2+2
    PK(NR)=PKR(N,NIT)
    ENDF
  ENDIF
100 CONTINUE
C
WRITE ('.') 'PK',PK(NR),R,R(NR)
100 CONTINUE
C
DO 200 L=1,(NRNG-1)
  IF (RANGE.GT.R(I).AND.RANGE.LT.R(I+1)) THEN
    ITS BETWEEN THESE TWO
    BOT=R(I)-R(I)
    TOP=RANGE-R(I)
    DELPK=PK(I+1)-PK(I)
    FRACPK=DELPK*TOP/BOT
    PKIT=FRACPK+PK(I)
  ENDF
200 CONTINUE
IF (RANGE.GT.R(NRNG)) PKIT=0.0
RETURN
END
C
SUBROUTINE TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,STIME)
  C
  C
  C
  INTERPOLATES MIN AND MAX ENGAGEMENT TIMES TO GIVE
  TIME IN ENGAGEMENT MODE
  C
  RNG=TRNG*1000.
  RMIN=0.0
  BOT=RMAX-RMIN
  TOP=RNG-RMIN
  DELT=TMAX-TMIN
  FRACT=DELT*TOP/BOT
  STIME=FRACT+TMIN
  RETURN
  END
C
SUBROUTINE VMASK (KSIDE,VEHICLE)

```

```

C FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C SO THAT MASKING CAN BE EVALUATED
C VEHICLE MASKING IS ONLY A POTENTIAL PROBLEM FOR VEHICLES
C WITHIN 100 METERS OF THE SHOOTER, SINCE TERRAIN DEVIATIONS
C WILL MAKE VEHICLE MASKING LARGELY
C IRRELEVANT NEAR THE TARGET AND OVER THE INTERVENING DISTANCES.

```

```

C COMMON/MASKV/VMSK(400,2)
C COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
C IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NH
C COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),IRSA(200,2),
C L1BM,L2M,L3M,L1RM
C CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C CHARACTER*16 BHIER,RHIER,IHIER
C CHARACTER*24 BUNIT,RUNIT

```

```

C KSIDE - 1 - BLUE 2 - RED
C IVEHICLE - VEHICLE WHICH IS SHOOTING
C VMSK(I,1) -- RANGE TO EACH VEHICLE IN FORCE STRUCTURE
C VMSK(I,2) -- ANGLE TO EACH VEHICLE IN FORCE STRUCTURE

```

```

C PL=3.14159
C INITIALIZE ARRAYS
C DO 10 I=1,400
C VMSK(I,1)=1.0
C VMSK(I,2)=0.0

```

```

10 CONTINUE
C FIND SHOOTING VEHICLE'S LOCATION
C IF (KSIDE.EQ.1) THEN
C XS=BXYA(IVEHICLE,1)
C YS=BXYA(IVEHICLE,2)
C ELSE
C XS=RXYA(IVEHICLE,1)
C YS=RXYA(IVEHICLE,2)
C ENDIF

```

```

C CYCLE THROUGH ALL BLUE AND RED VEHICLES TO FILL MASKING ARRAY
C IALL=NB+NR
C DO 20 I=1,IALL
C IF (LGT.NB) THEN
C IWANT=I-NB
C XT=RXYA(IWANT,1)
C YT=RXYA(IWANT,2)
C ELSE
C IWANT=I
C XT=BXYA(IWANT,1)
C YT=BXYA(IWANT,2)
C ENDIF

```

```

C RANGE=((XT-XS)**2+(YT-YS)**2)**.5
C IF (RANGE.GT.0.1) GO TO 20
C FIND ANGLE TO THIS TARGET
C TOP=YT-YS
C BOT=XT-XS
C IF (BOT.EQ.0.0) BOT=BOT+.001
C DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
C IF (BOT.LT.0.0) THEN
C IF (TOP.LT.0.0) DIR=DIR+180.
C IF (TOP.GT.0.0) DIR=180.-DIR
C ELSE
C IF (TOP.LT.0.0) DIR=360.-DIR
C ENDIF

```

```

C CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
C CALL FAZ (DIR,AZIN)
C VMSK(I,1)=RANGE
C VMSK(I,2)=AZIN
20 CONTINUE
C RETURN
C END

```

```

C SUBROUTINE VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C EVALUATES VEHICLE MASKING TO SEE IF TARGET CAN BE SEEN
C COMMON/MASKV/VMSK(400,2)

```

```

C MYSIDE SHOOTING VEHICLE SIDE (1 - BLUE, 2 - RED)
C K SHOOTING VEHICLE NUMBER
C J THE TARGET (IT IS OF COURSE IN THE FIELD OF VIEW)
C RANGE TARGET RANGE (DON'T LOOK FOR MASKING BEYOND THIS RANGE)
C AZIN AZIMUTH TO TARGET
C JSKIP FLAG TO SKIP THIS TARGET IF MASKED (-0 TO SKIP)
C VMSK VEHICLE RANGE AND AZIMUTH DATA FROM SHOOTING VEHICLE

```

```

C IF (MYSIDE.EQ.1) THEN
C ITSKIP=J
C MESKIP=K
C ELSE
C ITSKIP=J+NB
C MESKIP=K+NB
C ENDIF
C IALL=NB+NR
C PL=3.14159
C AVERAGE WIDTH OF AN ARMORED VEHICLE
C WIDTH=4/1000.
C DO 10 I=1,IALL
C IF (JSKIP.EQ.0) GO TO 10
C IF (LEQ.ITSKIP) GO TO 10
C IF (LEQ.MESKIP) GO TO 10
C IF (VMSK(I,1).GT.0.1) GO TO 10
C AZLFT=VMSK(I,2)-180./PI*ATAN(WIDTH/2/VMSK(I,1))
C AZRGT=VMSK(I,2)+180./PI*ATAN(WIDTH/2/VMSK(I,1))

```

```

C IF (MYSIDE.EQ.0) GO TO 10
C IF (LEQ.ITSKIP) GO TO 10
C IF (LEQ.MESKIP) GO TO 10
C IF (VMSK(I,1).GT.0.1) GO TO 10
C AZLFT=VMSK(I,2)-180./PI*ATAN(WIDTH/2/VMSK(I,1))
C AZRGT=VMSK(I,2)+180./PI*ATAN(WIDTH/2/VMSK(I,1))

```

```

IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.
IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.
IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.
IF (AZIN.GE.AZLFT.AND.AZIN.LE.AZRGT) JSKIP=0
10 CONTINUE
C RETURN
C END

```

```

C SUBROUTINE TERRMASK (XS,YS,XT,YT,JSKIP)
C EVALUATES TERRAIN MASKING BETWEEN SHOOTER AND TARGET.
C TO ENSURE EQUIVALENT TERRAIN MASKING CALCULATIONS BETWEEN
C RED AND BLUE OVER THE SAME TERRAIN, MASKING IS ALWAYS
C CALCULATED FROM BLUE TO RED, EVEN IF RED IS SHOOTER

```

```

C XS,YS SHOOTER REAL WORLD COORDINATES
C XT,YT TARGET REAL WORLD COORDINATES
C JSKIP FLAG TO SKIP THIS TARGET IF MASKED (-0 TO SKIP)

```

```

C COMMON/ELEVA/CONT(20),IELEV(10000)
C COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT

```

```

C DOUBLE PRECISION ANGLE,ANGLEI

```

```

C JSKIP=1

```

```

C WRITE (10,*) 'ANGLE, ANGLEI'

```

```

C FIND SHOOTER AND TARGET GRID COORDINATES
C TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C CENTER TO CENTER ABOUT THE SCALE.

```

```

C IXS=NINT(XS/SCALE)
C IYS=NINT(YS/SCALE)
C IXT=NINT(XT/SCALE)
C IYT=NINT(YT/SCALE)
C IF (IXS.LT.1) IXS=1
C IF (IXT.LT.1) IXT=1
C IF (IYS.LT.1) IYS=1
C IF (IYT.LT.1) IYT=1
C IF (IXS.GT.NX) IXS=NX
C IF (IXT.GT.NX) IXT=NX
C IF (IYS.GT.NY) IYS=NY
C IF (IYT.GT.NY) IYT=NY
C X1=IXS*SCALE
C X2=IXT*SCALE
C Y1=IYS*SCALE
C Y2=IYT*SCALE
C RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5

```

```

C FIND SHOOTER AND TARGET ELEVATION
C IS=NY*(IXS-1)+IYS
C SELEV=CONT(IELEV(IS))
C IT=NY*(IXT-1)+IYT
C TELEV=CONT(IELEV(IT))

```

```

C ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C FOR VEHICLE HEIGHT
C SELEV=SELEV+.002
C TELEV=TELEV+.002

```

```

C FIND ELEVATION ANGLE TO THIS TARGET
C TOP=TELEV-SELEV
C BOT=RANGE
C ANGLE=ATAN(TOP/BOT)

```

```

C TO EVALUATE MASKING, FIND THE ELEVATION ANGLE AT SCALE
C INCREMENTS ALONG THE LINE OF SIGHT. IF ELEVATION ANGLE IS GREATER
C THAN THIS ANGLE, SKIP THE TARGET, SINCE IT IS MASKED BY
C TERRAIN

```

```

C DY=Y2-Y1
C DX=X2-X1

```

```

C IF (ABS(DY).GT.ABS(DX)) THEN
C STEP THROUGH GRID ALONG Y AXIS
C SLOPE=(X2-X1)/(Y2-Y1)
C B=X2-SLOPE*Y2
C YL=Y1

```

```

20 CONTINUE
C IF (JSKIP.EQ.0) GO TO 200
C YL=Y1+SCALE*DY/ABS(DY)
C XL=B+SLOPE*YL
C RI=((Y1-Y1)**2+(X1-X1)**2)**.5
C IF (RI.GE.RANGE) GO TO 200
C IXI=NINT(XI/SCALE)
C IYI=NINT(YI/SCALE)
C ITI=NY*(IXI-1)+IYI
C XIXI=IXI*SCALE
C YIYI=IYI*SCALE
C RI=((YIYI-Y1)**2+(XIXI-X1)**2)**.5
C SELEVI=CONT(IELEV(ITI))
C FIND ELEVATION ANGLE TO THIS GRID SQUARE
C TOP=SELEVI-SELEV
C BOT=RI
C ANGLEI=ATAN(TOP/BOT)

```

```

C WRITE (10,*) 1,ANGLE,ANGLEI
C IF (ANGLEI.GT.ANGLE) JSKIP=0
C GO TO 20

```

```

200 CONTINUE
ELSE
C STEP THROUGH GRID ALONG X AXIS
SLOPE=(Y2-Y1)/(X2-X1)
B=Y2-SLOPE*X2
X1=X2
10 CONTINUE
IF (JSKIP.EQ.0) GO TO 100
X1=X1-SCALE*DX/ABS(DX)
Y1=B-SLOPE*X1
R1=(Y1-Y2)**2+(X1-X2)**2** .5
IF (R1.GE.RANGE) GO TO 100
DX=NINT(X1/SCALE)
Y1=NINT(Y1/SCALE)
IT=NINT(DX-1)/Y1
X1=X1-DX*SCALE
Y1=Y1-IT*SCALE
R1=(Y1-Y2)**2+(X1-X1)**2** .5
SELEVL=CONT(ELEV(IT))
C FIND ELEVATION ANGLE TO THIS GRID SQUARE
TOP=SELEVL-SELEV
BOT=R1
ANGLE=ATAN(TOP/BOT)
C WRITE (10,*) 2,ANGLE,ANGLE!
IF (ANGLE.LT.ANGLE) JSKIP=0
GO TO 10
100 CONTINUE
ENDIF
RETURN
END

SUBROUTINE BZOOM
C ZOOM WINDOW FOR BATTLE PHASE
C ZOOMS IN ON ACTIVE GRID
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/OLDMOVE/BXYOLD(200,2),RXYOLD(200,2)
CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

BLX BOTTOM LEFT X COORDINATE
BLY BOTTOM LEFT Y COORDINATE
XLEN ZOOM WINDOW SIDE LENGTH
ZFACTN NEW ZOOM FACTOR TO INCREASE SCALE

XUNIT=11.0/NX*SCFACT
YUNIT=.8/NY
XMIN=1.0E8
XMAX=0.0
YMIN=1.0E8
YMAX=0.0
DO 10 I=1,NB
IF (IRSYS(I,1).GT.10) GO TO 10
IF (BXYA(I,1).LT.XMIN) XMIN=BXYA(I,1)
IF (BXYA(I,1).GT.XMAX) XMAX=BXYA(I,1)
IF (BXYA(I,2).LT.YMIN) YMIN=BXYA(I,2)
IF (BXYA(I,2).GT.YMAX) YMAX=BXYA(I,2)
10 CONTINUE
DO 11 I=1,NR
IF (IRSYS(I,1).GT.10) GO TO 11
IF (RXYA(I,1).LT.XMIN) XMIN=RXYA(I,1)
IF (RXYA(I,1).GT.XMAX) XMAX=RXYA(I,1)
IF (RXYA(I,2).LT.YMIN) YMIN=RXYA(I,2)
IF (RXYA(I,2).GT.YMAX) YMAX=RXYA(I,2)
11 CONTINUE
XMIN=XMIN-SCALE
XMAX=XMAX+SCALE
XAVG=(XMIN+XMAX)/2
DX=XMAX-XMIN
YMIN=YMIN-SCALE
YMAX=YMAX+SCALE
YAVG=(YMIN+YMAX)/2
DY=YMAX-YMIN
C WINDOW EDGE LENGTH IN GRID UNITS
IF (DY.GT.DX) THEN
XLEN=NINT(DY/SCALE)
IF (XLEN.LT.4.1) XLEN=4.1
XMIN=XAVG-DY/2
YMIN=YAVG-DY/2
ELSE
XLEN=NINT(DX/SCALE)
IF (XLEN.LT.4.1) XLEN=4.1
XMIN=XAVG-DX/2
YMIN=YAVG-DX/2

```

```

ENDIF
GRID NX,NY OF WINDOW BOTTOM LEFT CORNER
BLX=NINT(XMIN/SCALE)*XUNIT
BLY=NINT(YMIN/SCALE)*YUNIT
C CALCULATE NEW ZOOM FACTOR
XLEN1=X*XUNIT
ZFACTN=XLEN1/XLEN/XUNIT*ZFACT
ZFACTN=ZFACT
XMAX=XLEN*XUNIT
YMAX=XLEN*YUNIT
RETURN
END

SUBROUTINE EUNITS
C ERASES UNIT LOCATIONS (VEHICLES) ON MAP
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/OLDMOVE/BXYOLD(200,2),RXYOLD(200,2)
CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*1 BLANK
CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
DIMENSION XARRAY(4),YARRAY(4)

ISKIP=0
C DRAWS AND FILLS A SQUARE FOR EACH UNIT LOCATION
C SIDE LENGTH IS 5 METERS SINCE LARGEST UNIT IS A SQUAD VEHICLE
C WHEN AT MAX ZOOM
C AT MIN ZOOM, BOX IS DRAWN LARGER SO YOU CAN SEE IT
C CALL PLOTS (0,1,0)
C CALL FACTOR (ZFACT)
C XUNIT=11.0/NX*SCFACT
C YUNIT=.8/NY
C GXUNIT=11.0/NX*SCFACT
C GYUNIT=.8/NY
C XUNIT=GXUNIT/SCALE
C YUNIT=GYUNIT/SCALE
C XL=XUNIT*.005/SCALE*2
C YL=YUNIT*.005/SCALE*2
C XL=XUNIT*.150/SCALE
C YL=YUNIT*.150/SCALE
C XL=XUNIT*.150/SCALE
C YL=YUNIT*.150/SCALE
C IF (XLEN.GT.0) THEN
C CALL FACTOR (ZFACTN)
C XL=XUNIT*.150/SCALE*ZFACT/ZFACTN
C YL=YUNIT*.150/SCALE*ZFACT/ZFACTN
C ENDF
C DO 200 N=1,2
C DO 100 I=1,200
C ISKIP=0
C IF (N.EQ.1) THEN
C IHIER=BHIER(I)
C BLANK=BHIER(1:1)
C ELSE
C IHIER=RHIER(I)
C BLANK=BHIER(1:1)
C ENDF
C IF (BLANK.EQ.'B') GO TO 60
C IF (BLANK.EQ.'R') GO TO 60
C GO TO 100
50 IF (N.EQ.1) THEN
C X=BXYOLD(I,1)
C Y=BXYOLD(I,2)
C CALL NEWPEN(0)
C IF (IRSYS(I,2).EQ.0) CALL NEWPEN(0)
C ELSE
C X=RXYOLD(I,1)
C Y=RXYOLD(I,2)
C CALL NEWPEN(0)
C IF (IRSYS(I,2).EQ.0) CALL NEWPEN(0)
C ENDF
C X=X*XUNIT
C Y=Y*YUNIT
C IF (X.LE.0.0) ISKIP=1
C IF (Y.LE.0.0) ISKIP=1
C IF (XLEN.GT.0) THEN
C CALL ZOOMIT (X,Y,ISKIP)
C IF (Y.GT.(.95*YMAX)) ISKIP=1
C IF (X.LT.(.05*XMAX)) ISKIP=1
C ENDF
C IF (ISKIP.EQ.1) GO TO 100
C DRAW AND FILL FOUR SIDED SQUARE
C XARRAY(1)=X-XL/2
C XARRAY(2)=X-XL/2
C XARRAY(3)=X-XL/2
C XARRAY(4)=X-XL/2
C YARRAY(1)=Y-YL/2
C YARRAY(2)=Y-YL/2
C YARRAY(3)=Y-YL/2
C YARRAY(4)=Y-YL/2

```



```

ENDIF
IF (ISKP.EQ.1) GO TO 100
C DRAW CIRCLE
CALL CIRCLE (X,Y,RADIUS)
IEERASE=IEERASE+1
PTERASE(IEERASE,1)=PTENG(1,1)
PTERASE(IEERASE,2)=PTENG(1,2)
PTERASE(IEERASE,3)=PTENG(1,3)
100 CONTINUE
RETURN
END

SUBROUTINE MPUFFS
MOVES OBSCURANT PUFFS
COMMON/PUFFS/WSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENOPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
COMMON/BATTLE/ITIME,ITGTR(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
C PL=3.14159
DT=DTPUFF
V=WSPEED*1.853
AZIN=IWDIR
CALL FDIR(AZIN,AZOUT)

IF (NDUST.GT.0) THEN
MOVE DUST
DO 100 I=IDUST1,IDUST2
MOVE IT
PTDUST(I,1)=PTDUST(I,1)+V*DT*COS(AZOUT*PI/180)/3600.
PTDUST(I,2)=PTDUST(I,2)+V*DT*SIN(AZOUT*PI/180)/3600.
PTDUST(I,3)=PTDUST(I,3)+DTPUFF
100 CONTINUE
ENDIF

IF (NENG.GT.0) THEN
MOVE ENGINE SMOKE
DO 200 I=IENG1,IENG2
MOVE IT
PTENG(I,1)=PTENG(I,1)+V*DT*COS(AZOUT*PI/180)/3600.
PTENG(I,2)=PTENG(I,2)+V*DT*SIN(AZOUT*PI/180)/3600.
PTENG(I,3)=PTENG(I,3)+DTPUFF
200 CONTINUE
ENDIF

IF (NSMK.NE.0) THEN
MOVE SMOKE GRENADE CURTAIN
DO 300 I=ISMK1,ISMK2
MOVE IT
PSMK(I,1)=PSMK(I,1)+V*DT*COS(AZOUT*PI/180)/3600.
PSMK(I,2)=PSMK(I,2)+V*DT*SIN(AZOUT*PI/180)/3600.
TASMK(I,1)=TASMK(I,1)+DTPUFF
300 CONTINUE
ENDIF

IF (NART.NE.0) THEN
MOVE ARTILLERY SMOKE CURTAIN
DO 400 I=IART1,IART2
MOVE IT
PART(I,1)=PART(I,1)+V*DT*COS(AZOUT*PI/180)/3600.
PART(I,2)=PART(I,2)+V*DT*SIN(AZOUT*PI/180)/3600.
TAART(I,1)=TAART(I,1)+DTPUFF
400 CONTINUE
ENDIF
RETURN
END

SUBROUTINE DELPUFFS
DELETE DISSIPATED OBSCURANTS
COMMON/PUFFS/WSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENOPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
COMMON/BATTLE/ITIME,ITGTR(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
C IF (NDUST.GT.0) THEN
C DELETE DISSIPATED DUST
DO 100 I=IDUST1,IDUST2
J=I
IF (PTDUST(J,3).GE.TENDPUFF) THEN
THIS PUFF HAS EXPIRED
ISTART=J+1
PTDUST(J,1)=1
PTDUST(J,2)=1
PTDUST(J,3)=1.E6
NDUST=NDUST-1
ELSE
THIS IS THE FIRST PUFF TO CARRY FORWARD
ISTART=J
GO TO 101
ENDIF
100 CONTINUE

101 CONTINUE
IDUST1=ISTART
IF (NDUST.LE.0.OR.IDUST1.GE.IDUST2) THEN
IDUST1=0
IDUST2=0
NDUST=0
ELSE
NDUST=IDUST2-IDUST1+1
ENDIF
ENDIF

IF (NENG.GT.0) THEN
DELETE DISSIPATED ENGINE SMOKE
DO 200 I=IENG1,IENG2
J=I
IF (PTENG(J,3).GE.TENOPUFF) THEN
THIS PUFF HAS EXPIRED
ISTART=J+1
PTENG(J,1)=1
PTENG(J,2)=1
PTENG(J,3)=1.E6
NENG=NENG-1
ELSE
THIS IS THE FIRST PUFF TO CARRY FORWARD
ISTART=J
GO TO 201
ENDIF
200 CONTINUE
201 CONTINUE
IENG1=ISTART
IF (NENG.LE.0.OR.IENG1.GE.IENG2) THEN
IENG1=0
IENG2=0
NENG=0
ELSE
NENG=IENG2-IENG1+1
ENDIF
ENDIF

IF (NSMK.NE.0) THEN
DELETE DISSIPATED SMOKE GRENADE CURTAIN
DO 300 I=ISMK1,ISMK2
J=I
IF (TASMK(J,1).GE.TENOPUFF) THEN
THIS PUFF HAS EXPIRED
ISTART=J+1
ELSE
THIS IS THE FIRST PUFF TO CARRY FORWARD
ISTART=J
GO TO 301
ENDIF
300 CONTINUE
301 CONTINUE
ISMK1=ISTART
IF (ISMK1.GE.ISMK2) THEN
ISMK1=0
ISMK2=0
NSMK=0
ELSE
NSMK=ISMK2-ISMK1+1
ENDIF
ENDIF

IF (NART.NE.0) THEN
DELETE DISSIPATED ARTILLERY SMOKE PUFF
DO 400 I=IART1,IART2
J=I
IF (TAART(J,1).LE.0.0) THEN
THIS PUFF HAS EXPIRED
ISTART=J+1
ELSE
THIS IS THE FIRST PUFF TO CARRY FORWARD
ISTART=J
GO TO 401
ENDIF
400 CONTINUE
401 CONTINUE
IART1=ISTART
IF (IART1.GE.IART2) THEN
IART1=0
IART2=0
NART=0
ELSE
NART=IART2-IART1+1
ENDIF
ENDIF
RETURN
END

SUBROUTINE NEWPUFFS
CREATES NEW OBSCURANT PUFFS AS REQUIRED (DUST AND ENGINE)
COMMON/PUFFS/WSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENOPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
COMMON/FORCE/BIHER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IBSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM

```

```

COMMON/OBSOLETE/NGRENS(200,2),NGREN(200,2),IBESMK(200),
& IRESMK(200)
CHARACTER*16 BHIER,RHIER,NIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2ML,L3M,L1RM
C
IF (IDUST.EQ.1) THEN
C PLAYING DUST, CREATE NEW DUST PUFFS
IF (NOUST.GT.0) THEN
C TEST TO SEE IF DUST ARRAY NEEDS TO BE COMPRESSED BACK
C TO COLUMN 1
ITEST=NR+NB+IDUST2
IF (ITEST.GT.NOUSTMX) THEN
C COMPRESS ARRAY BEFORE STARTING
J=0
DO 100 I=IDUST1,IDUST2
J=J+1
PTDUST(J,1)=PTDUST(L1)
PTDUST(J,2)=PTDUST(L2)
PTDUST(J,3)=PTDUST(L3)
100 CONTINUE
IDUST1=1
IDUST2=J
ENDIF
C CREATE NEW PUFFS AFTER IDUST2
J=0
DO 200 I=1,NB
IF (IBSYS(L2).EQ.0) GO TO 200
C SKIP KILLED VEHICLES
IF (IRSA(L1).LT.0) GO TO 200
C SKIP VEHICLES MOVING UNDER 6 KM/HR
J=J+1
X=BXA(L1)
Y=BYA(L1)
K=IDUST2+J
PTDUST(K,1)=X
PTDUST(K,2)=Y
PTDUST(K,3)=DTPUFF
C WRITE (,"") PTDUST 1 2 3
C WRITE (,"") (PTDUST(K,L),L=1,3)
200 CONTINUE
DO 300 I=1,NR
IF (IRSYS(L2).EQ.0) GO TO 300
C SKIP KILLED VEHICLES
IF (IRSA(L1).LT.0) GO TO 300
C SKIP VEHICLES MOVING UNDER 6 KM/HR
J=J+1
X=BXA(L1)
Y=BYA(L1)
K=IDUST2+J
PTDUST(K,1)=X
PTDUST(K,2)=Y
PTDUST(K,3)=DTPUFF
300 CONTINUE
IDUST2=IDUST2+J
NOUST=IDUST2-IDUST1+1
ENDIF
C
C ENGINE SMOKE
IF (NENG.GT.0) THEN
C TEST TO SEE IF ENGINE SMOKE ARRAY NEEDS TO BE COMPRESSED BACK
C TO COLUMN 1
ITEST=NR+NB+IENG2
IF (ITEST.GT.NENGMX) THEN
C COMPRESS ARRAY BEFORE STARTING
J=0
DO 110 I=IENG1,IENG2
J=J+1
PTENG(J,1)=PTENG(L1)
PTENG(J,2)=PTENG(L2)
PTENG(J,3)=PTENG(L3)
110 CONTINUE
IENG1=1
IENG2=J
ENDIF
C CREATE NEW ENGINE SMOKE PUFFS AFTER IENG2
J=0
DO 210 I=1,NB
IF (IBSYS(L2).EQ.0) GO TO 210
C SKIP KILLED VEHICLES
IF (IBESMK(L1).EQ.0) GO TO 210
C SKIP VEHICLES WITH ENGINE SMOKE TURNED OFF
J=J+1
X=BXA(L1)
Y=BYA(L1)
K=IENG2+J
PTENG(K,1)=X
PTENG(K,2)=Y
PTENG(K,3)=DTPUFF
C WRITE (,"") PTDUST 1 2 3
C WRITE (,"") (PTDUST(K,L),L=1,3)
210 CONTINUE
DO 310 I=1,NR
IF (IRSYS(L2).EQ.0) GO TO 310
C SKIP KILLED VEHICLES
IF (IRESMK(L1).EQ.0) GO TO 310

```

```

C SKIP VEHICLES WITH ENGINE SMOKE TURNED OFF
J=J+1
X=BXA(L1)
Y=BYA(L1)
K=IENG2+J
PTENG(K,1)=X
PTENG(K,2)=Y
PTENG(K,3)=DTPUFF
310 CONTINUE
IENG2=IENG2+J
NENG=IENG2-IENG1+1
C
RETURN
END
C
SUBROUTINE ERASEPUFFS
ERASES ALL PREVIOUSLY DRAWN DUST AND ENGINE SMOKE PUFFS
COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
& IDUST1,IDUST2,NOUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/EDPUFFS/IDERASE,PTDRASE(2400,3)
COMMON/ENPUFFS/IEERASE,PTERASE(2400,3)
C
C ISKIP=0
C
C DRAWS A BLACK CIRCLE FOR EACH PUFF LOCATION.
C RADIUS IS BASED ON PUFF TIME WHEN AT MAX ZOOM.
C AT MIN ZOOM, CIRCLE IS DRAWN LARGER SO YOU CAN SEE IT.
C
CALL FACTOR (ZFACT)
GXUNIT=.11,GNX*SCFACT
GYUNIT=.8,GNY
XUNIT=GXUNIT/SCALE
YUNIT=GYUNIT/SCALE
DO 200 J=1,2
IF (J.EQ.1) IERASE=IDERASE
IF (J.EQ.2) IERASE=IEERASE
DO 100 I=1,IERASE
IF (J.EQ.1) THEN
TIME=PTDRASE(L3)
X=PTDRASE(L1)*XUNIT
Y=PTDRASE(L2)*YUNIT
ELSE
TIME=PTERASE(L3)
X=PTERASE(L1)*XUNIT
Y=PTERASE(L2)*YUNIT
ENDIF
RADIUS=TIME*EXPUFF*XUNIT*1.0
C WRITE (,"") RADIUS X Y
C WRITE (,"") RADIUS X Y
IF (XLEN.GT.0) THEN
CALL FACTOR (ZFACTN)
RADIUS=RADIUS*ZFACT/ZFACTN
ENDIF
CALL NEWPEN (0)
IF ((X-RADIUS).LE.0.0) ISKIP=1
IF ((Y-RADIUS).LE.0.0) ISKIP=1
IF (XLEN.GT.0) THEN
CALL ZOOMIT (X,Y,ISKIP)
IF ((Y-RADIUS).GT.(.95*YMAX)) ISKIP=1
ENDIF
C IF (ISKIP.EQ.1) GO TO 100
C DRAW CIRCLE
CALL CIRCLE (X,Y,RADIUS)
100 CONTINUE
200 CONTINUE
RETURN
END
C
SUBROUTINE OBSCURANTS
C
C ENTER ORDERS FOR UNITS TO EMPLOY ENGINE SMOKE, SMOKE GRENADES
C OR ARTILLERY SMOKE (NOT DONE YET)
C
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/TOURS/X1(1100),Y1(1100),X2(1100),Y2(1100),JC(1100),
& NC(15),NCOL(15),JCON
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXA(200,2),BYA(200,2),IRSA(200,2),IRSA(200,2),
& L1BM,L2ML,L3M,L1RM
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRNB(30),NRNR(30),TEMAB(30),TEMAR(30),TEMINB(30),
& TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MILB(10,3),MILR(10,3)
COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& BXYP(200,2),RXYP(200,2),ISPOB(200),ISPOR(200)
COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
& IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)

```

```

COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
& TRMFR(200),TRDES(200),TRDES(200),
COMMON/BATTLE/ITIME,ITIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200),
COMMON/OBSOURE/NGREN(200,2),NGREN(200,2),IBESMK(200),
& IRESMK(200),
COMMON/TYPES/NSAT,NCOMP,ICOMP(10),NPLAT(10),IPLAT(10,10),
& NSOD(10,10),JPLAT,JSD
C CHARACTER*8 BFNAME,RFNAME
C CHARACTER*90 LINE
C CHARACTER*18 BHIER,RHIER,IHIER
C CHARACTER*24 BUNIT,RUNIT
C CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C CHARACTER*1 C2
C CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
C KSIDE=0
10 WRITE(' ')
WRITE(' ') 'OBSOURE ORDERS'
WRITE(' ') 'enter BLUE or RED, or QUIT'
WRITE(' ')
READ(' ',ERR=10) C2
IF (C2(1:1).EQ.'B') KSIDE=1
IF (C2(1:1).EQ.'R') KSIDE=2
IF (C2(1:1).EQ.'Q') GO TO 1000
IF (KSIDE.EQ.0) GO TO 10
C
5 WRITE(' ')
WRITE(' ') 'CURRENT STATUS'
11 FORMAT(1X,A90)
C IF (C2(1:1).EQ.'B') WRITE(' ',11) L1BM
C IF (C2(1:1).EQ.'R') WRITE(' ',11) L1RM
WRITE(' ',11) 'UNIT HIERARCHY' 'IDENTITY' 'ENGINE'
& SMOKE 'SMOKE GRENADES'
WRITE(' ')
& 0 = OFF # SHOTS FIRE=1' 1 = ON
IF (C2(1:1).EQ.'B') THEN
C BLUE
KK=0
DO 130 K=1,NB
121 FORMAT(1X,3,1X,A16,A24,3(16,4X))
IF (IBSYS(K,2).EQ.1.AND.IBSYS(K,1).LE.10) THEN
WRITE(' ',121) K,BHIER(K),BUNIT(K),IBESMK(K),NGREN(1),
& NGREN(2),
KK=KK+1
ENDIF
IF (KK.EQ.20) THEN
READ(' ')
KK=0
ENDIF
130 CONTINUE
C
ELSE
RED
KK=0
DO 140 K=1,NR
IF (IRSYS(K,2).EQ.1.AND.IRSYS(K,1).LE.10) THEN
WRITE(' ',121) K,RHIER(K),RUNIT(K),IRESMK(K),NGREN(1),
& NGREN(2),
KK=KK+1
ENDIF
IF (KK.EQ.20) THEN
READ(' ')
KK=0
ENDIF
140 CONTINUE
ENDIF
141 WRITE(' ') 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
WRITE(' ')
READ(' ',ERR=141) ICOMMAND
IF (ICOMMAND.EQ.0) GO TO 5
IF (ICOMMAND.EQ.-1) GO TO 10
C
142 WRITE(' ') 'ENTER NUMBER RANGE OF SUBORDINATES FOR'
WRITE(' ') 'SUB-UNIT ALLOCATION (#START #END) (0 0 FOR NONE)'
WRITE(' ')
READ(' ',ERR=142) NSUB1,NSUB2
NSUB=1
IF (NSUB1.LE.0) THEN
NSUB1=0
NSUB2=0
NSUB=0
ENDIF
IF (NSUB2.LE.0) THEN
NSUB1=0
NSUB2=0
NSUB=0
ENDIF
ENDIF
IF (NSUB.EQ.0) GO TO 145
C
C143 WRITE(' ') 'SUB-UNIT ALLOCATIONS'
ISUB=2
C
145 WRITE(' ')
WRITE(' ') 'ENTER 1 TO TURN ENGINE SMOKE ON; 0 -- OFF'
WRITE(' ')

```

```

READ(' ',ERR=145) IENGSMK
IF (IENGSMK.LT.0.OR.IENGSMK.GT.1) GO TO 146
146 WRITE(' ')
WRITE(' ') 'ENTER 1 TO FIRE SMOKE GRENADES; 0 -- NO FIRE'
WRITE(' ')
READ(' ',ERR=146) IGSMK
IF (IGSMK.LT.0.OR.IGSMK.GT.1) GO TO 146
C
C BLUE
IF (KSIDE.EQ.1) THEN
IF (IBSYS(ICOMMAND,2).EQ.1.AND.IBSYS(ICOMMAND,1).LE.10) THEN
IBESMK(ICOMMAND)=IENGSMK
IF (NGREN(ICOMMAND,1).GT.0) NGREN(ICOMMAND,2)=IGSMK
ENDIF
IF (ISUB.EQ.2) THEN
EVERY ONE GETS THE SAME ORDERS
DO 175 I=NSUB1,NSUB2
IF (IBSYS(I,2).EQ.1.AND.IBSYS(I,1).LE.10) THEN
IBESMK(I)=IENGSMK
IF (NGREN(I,1).GT.0) NGREN(I,2)=IGSMK
ENDIF
175 CONTINUE
ENDIF
C
C RED
IF (KSIDE.EQ.2) THEN
IF (IRSYS(ICOMMAND,2).EQ.1.AND.IRSYS(ICOMMAND,1).LE.10) THEN
IRESMK(ICOMMAND)=IENGSMK
IF (NGREN(ICOMMAND,1).GT.0) NGREN(ICOMMAND,2)=IGSMK
ENDIF
IF (ISUB.EQ.2) THEN
EVERY ONE GETS THE SAME ORDERS
DO 185 I=NSUB1,NSUB2
IF (IRSYS(I,2).EQ.1.AND.IRSYS(I,1).LE.10) THEN
IRESMK(I)=IENGSMK
IF (NGREN(I,1).GT.0) NGREN(I,2)=IGSMK
ENDIF
185 CONTINUE
ENDIF
C
GO TO 141
1000 CONTINUE
RETURN
END
SUBROUTINE GRENADES
C
C FIRES SMOKE GRENADES AND PUFFS CURTAINS
COMMON/PUFFS/IWSPEED,IWDIR,IOUST,EXPUFF,OTPUFF,TENDPUFF,
& IOUST1,IOUST2,NOUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NOUSTMX,NENGMX,NSMKMX,
& NARTMX
COMMON/OBSOURE/NGREN(200,2),NGREN(200,2),IBESMK(200),
& IRESMK(200),
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/BATTLE/ITIME,ITIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200),
COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
& IRSECOND(200,2),WRNG(200,2),WRNGR(200,2)
CHARACTER*18 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C
C SMOKE GRENADES
IF (NSMK.GT.0) THEN
C
C TEST TO SEE IF GRENADE SMOKE ARRAY NEEDS TO BE COMPRESSED BACK
C TO COLUMN 1
ITEST=NR+NB+ISMK2
IF (ITEST.GT.NSMKMX) THEN
COMPRESS ARRAY BEFORE STARTING
J=0
DO 110 I=ISMK1,ISMK2
J=J+1
PSMK(J,1)=PSMK(I,1)
PSMK(J,2)=PSMK(I,2)
TASMK(J,1)=TASMK(I,1)
TASMK(J,2)=TASMK(I,2)
110 CONTINUE
ISMK1=1
ISMK2=J
ENDIF
C
C CREATE NEW SMOKE GRENADE CURTAINS AFTER ISMK2
J=0
P=3.14159
DO 210 I=1,NB
IF (IBSYS(I,2).EQ.0) GO TO 210
C
C SKIP KILLED VEHICLES
IF (NGREN(I,2).EQ.0) GO TO 210
C
C SKIP VEHICLES WITH GRENADE SMOKE TURNED OFF
IF (NGREN(I,1).EQ.0) GO TO 210
C
C SKIP VEHICLES THAT ARE OUT OF GRENADES

```



```

J=J+1
X=BX*YA(L1)
Y=BY*YA(L2)
K=ISMK2-J
C FIRE GRENADE AND REDUCE NUMBER BY 1
  NGREN(L1)=NGREN(L1)-1
  WRITE (10,*) 'BLUE', '1', 'FIRES SMOKE GRENADE VOLLEY'
C ORIENTATION OF SMOKE CURTAIN IS PERPENDICULAR TO THE CENTER
C OF THE PRIMARY SECTOR OF FIRE
  A1=ISPRIME(L1)
  A2=ISPRIME(L2)
  IF (A1.GT.A2) A2=A2-360.0
  AVG=A1+A2
  AVG=AVG/2.0
  APERP=AVG-90.0
  IF (APERP.GE.360.0) APERP=APERP-360.0
  TASM(K,2)=APERP
C CURTAIN IS THROWN A DISTANCE OF 20 METERS IN FRONT OF THE
C VEHICLE
  CALL FDIR (AVG,AZIN)
  PSM(K,1)=X-COS(AZIN/180.*PI)*.02
  PSM(K,2)=Y-SIN(AZIN/180.*PI)*.02
  TASM(K,1)=DTPUFF
210 CONTINUE

```

```

DO 310 I=1,NR
  IF (IRSY9(L2).EQ.0) GO TO 310
  SKIP KILLED VEHICLES
  IF (NGREN(L2).EQ.0) GO TO 310
  SKIP VEHICLES WITH GRENADE SMOKE TURNED OFF
  IF (NGREN(L1).EQ.0) GO TO 310
  SKIP VEHICLES THAT ARE OUT OF GRENADES
  J=J+1
  X=BX*YA(L1)
  Y=BY*YA(L2)
  K=ISMK2-J
C FIRE GRENADE AND REDUCE NUMBER BY 1
  NGREN(L1)=NGREN(L1)-1
  WRITE (10,*) 'RED', '1', 'FIRES SMOKE GRENADE VOLLEY'
C ORIENTATION OF SMOKE CURTAIN IS PERPENDICULAR TO THE CENTER
C OF THE PRIMARY SECTOR OF FIRE
  A1=ISPRIME(L1)
  A2=ISPRIME(L2)
  IF (A1.GT.A2) A2=A2-360.0
  AVG=A1+A2
  AVG=AVG/2.0
  APERP=AVG-90.0
  IF (APERP.GE.360.0) APERP=APERP-360.0
  TASM(K,2)=APERP
C CURTAIN IS THROWN A DISTANCE OF 20 METERS IN FRONT OF THE
C VEHICLE
  CALL FDIR (AVG,AZIN)
  PSM(K,1)=X-COS(AZIN/180.*PI)*.02
  PSM(K,2)=Y-SIN(AZIN/180.*PI)*.02
  TASM(K,2)=APERP
310 CONTINUE
  ISMK2=ISMK2-J
  NSMK=ISMK2-ISMK1+1

```

```

C RETURN
C END

SUBROUTINE PSMKCURTAINS
C PLOTS SMOKE GRENADE CURTAINS
  COMMON/PUFFS/WSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
  & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
  & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
  & TASM(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
  & NARTMX
  COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
  COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
  COMMON/EDSMKG/ISGERASE,PSGERASE(400,2),ASGERASE(400)

```

```

C ISKIP=0
C
C DRAWS A GREEN LINE FOR EACH SMOKE CURTAIN LOCATION.
C AND SAVES THIS LINE FOR LATER ERASING.
C

```

```

  CALL FACTOR (ZFACT)
  GXUNIT=11.0/NX*SCFACT
  GYUNIT=8.5/NY
  XUNIT=GXUNIT/SCALE
  YUNIT=GYUNIT/SCALE
  ISGERASE=0
C CURTAIN LENGTH IS 150 FEET, BUT PLOTTED AT TWICE SCALE (LIKE
C WITH VEHICLES SO THEY CAN BE SEEN BETTER)
  WCURT=150.*3048/1000.*2.0
  PI=3.14159
  CALL NEWPEN (10)

```

```

DO 100 I=ISMK1,ISMK2
  CENTER POINT OF CURTAIN
  XC=PSMK(I,1)
  YC=PSMK(I,2)
C CONVERT ORIENTATION AZIMUTH TO AN ANGLE
  CALL FDIR (TASM(I,2),AZIN)
C FIND STARTING POINT OF CURTAIN
  X1=XC-COS(AZIN/180.*PI)*WCURT/2.
  Y1=YC-SIN(AZIN/180.*PI)*WCURT/2.

```

```

C ENDIG POINT
  X2=XC-COS(AZIN/180.*PI)*WCURT/2.
  Y2=YC-SIN(AZIN/180.*PI)*WCURT/2.
C DRAW THIS LINE
  IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
  XS=X1*XUNIT
  YS=Y1*YUNIT
  ISKIP=0
  IF (XLEN.GT.0.) CALL ZOOMIT (XS,YS,ISKIP)
  IF (ISKIP.EQ.1) GO TO 100
  CALL PLOT (XS,YS,3)
  XT=X2*XUNIT
  YT=Y2*YUNIT
  IF (XLEN.GT.0.) CALL ZOOMIT (XT,YT,ISKIP)
  IF (ISKIP.EQ.1) GO TO 100
C DRAW LINE
  CALL PLOT (XT,YT,2)
  ISGERASE=ISGERASE+1
  PSGERASE(ISGERASE,1)=PSMK(I,1)
  PSGERASE(ISGERASE,2)=PSMK(I,2)
  ASGERASE(ISGERASE)=TASM(I,2)
100 CONTINUE
  RETURN
  END

```

```

SUBROUTINE ERSKMKCURT
  ERASES ALL PREVIOUSLY DRAWN SMOKE GRENADE CURTAINS
  COMMON/PUFFS/WSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
  & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
  & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
  & TASM(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
  & NARTMX
  COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
  COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
  COMMON/EDSMKG/ISGERASE,PSGERASE(400,2),ASGERASE(400)

```

```

  ISKIP=0
  C DRAWS A BLACK LINE FOR EACH SMOKE CURTAIN LOCATION.

```

```

  CALL FACTOR (ZFACT)
  GXUNIT=11.0/NX*SCFACT
  GYUNIT=8.5/NY
  XUNIT=GXUNIT/SCALE
  YUNIT=GYUNIT/SCALE

  DO 100 I=1,ISGERASE
  C CURTAIN LENGTH IS 150 FEET, BUT PLOTTED AT TWICE SCALE (LIKE
  C WITH VEHICLES SO THEY CAN BE SEEN BETTER)
  WCURT=150.*3048/1000.*2.0
  PI=3.14159
  CALL NEWPEN (10)
C CENTER POINT OF CURTAIN
  XC=PSGERASE(I,1)
  YC=PSGERASE(I,2)
C CONVERT ORIENTATION AZIMUTH TO AN ANGLE
  CALL FDIR (ASGERASE(I),AZIN)
C FIND STARTING POINT OF CURTAIN
  X1=XC-COS(AZIN/180.*PI)*WCURT/2.
  Y1=YC-SIN(AZIN/180.*PI)*WCURT/2.
C ENDING POINT
  X2=XC-COS(AZIN/180.*PI)*WCURT/2.
  Y2=YC-SIN(AZIN/180.*PI)*WCURT/2.
C DRAW THIS LINE
  IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
  XS=X1*XUNIT
  YS=Y1*YUNIT
  ISKIP=0
  IF (XLEN.GT.0.) CALL ZOOMIT (XS,YS,ISKIP)
  IF (ISKIP.EQ.1) GO TO 100
  CALL PLOT (XS,YS,3)
  XT=X2*XUNIT
  YT=Y2*YUNIT
  IF (XLEN.GT.0.) CALL ZOOMIT (XT,YT,ISKIP)
  IF (ISKIP.EQ.1) GO TO 100
C DRAW LINE
  CALL PLOT (XT,YT,2)
100 CONTINUE
  RETURN
  END

```

```

SUBROUTINE DMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C FIND ANGLE AND RANGE TO ALL DUST PUFFS WITHIN MAX RANGE OF
C THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
C
  COMMON/MASK/DMSK(2400,3),NDMSK
  COMMON/PUFFS/WSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
  & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
  & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
  & TASM(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
  & NARTMX

```

```

C XS,YS - SHOOTER COORDINATES
C RMAX1,2,3 - MAX RANGE OF SYSTEM WEAPONS
C DMSK(I,1) - X COORD OF EACH DUST PUFF
C DMSK(I,2) - Y COORD OF EACH DUST PUFF

```

```

C DMSK(L3) - RADIUS FOR EACH DUST PUFF
C NMSK - NUMBER OF DUST PUFFS WITHIN MAX RANGE
C
PL=3.14159
NMSK=0
IF (NDUST.GT.0) THEN
C CYCLE THROUGH ALL DUST PUFFS TO FILL MASKING ARRAY
RMAXX=0
IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
RMAXX=RMAXX/1000.0
DO 20 I=IDUST1,IDUST2
XD=PTDUST(L1)
YD=PTDUST(L2)
RADIUS=PTDUST(L3)*EXPUFF
RANGE=((XD-XS)**2+(YD-YS)**2)**.5
IF (RANGE-RADIUS).GT.RMAXX GO TO 20
NMSK=NMSK+1
FIND ANGLE TO THIS DUST PUFF
TOP=YD-YS
BOT=XD-XS
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
IF (BOT.LT.0.0) THEN
IF (TOP.LT.0.0) DIR=DIR+180.
IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
IF (TOP.LT.0.0) DIR=360.-DIR
ENDIF
CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (DIR,AZIN)
DMSK(NMSK,1)=XD
DMSK(NMSK,2)=YD
DMSK(NMSK,3)=RADIUS
20 CONTINUE
ENDIF
RETURN
END

SUBROUTINE DSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
EVALUATES DUST MASKING BETWEEN SHOOTER AND TARGET
C
C XS,YS SHOOTER REAL WORLD COORDINATES
C XT,YT TARGET REAL WORLD COORDINATES
C AZIN AZIMUTH TO TARGET
C JSKIP FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)
C
COMMON/ELEVA/CONT(20),IELEV(10000)
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/MASK/DMSK(2400,3),NMSK
C
DOUBLE PRECISION ANGLE,ANGLEI
C
PL=3.14159
C
WRITE (10,*) 'NMSK',NMSK
IF (NMSK.GT.0) THEN
C
C FIND SHOOTER AND TARGET GRID COORDINATES
C TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C CENTER TO CENTER ABOUT THE SCALE.
RANGE=((YT-YS)**2+(XT-XS)**2)**.5
IXS=NINT(XS/SCALE)
IYS=NINT(YS/SCALE)
IXT=NINT(XT/SCALE)
IYT=NINT(YT/SCALE)
IF (IXS.LT.1) IXS=1
IF (IXT.LT.1) IXT=1
IF (IYS.LT.1) IYS=1
IF (IYT.LT.1) IYT=1
IF (IXS.GT.NX) IXS=NX
IF (IXT.GT.NX) IXT=NX
IF (IYS.GT.NY) IYS=NY
IF (IYT.GT.NY) IYT=NY
X1=IXS*SCALE
X2=IXT*SCALE
Y1=IYS*SCALE
Y2=IYT*SCALE
RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5
C
C FIND SHOOTER AND TARGET ELEVATION
IS=NY*(IXS-1)+IYS
SELEV=CONT(IELEV(IS))
IT=NY*(IXT-1)+IYT
TELEV=CONT(IELEV(IT))
C
C ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C FOR VEHICLE HEIGHT
SELEV=SELEV+.002
TELEV=TELEV+.002
C
C FIND ELEVATION ANGLE TO THIS TARGET
TOP=TELEV-SELEV
BOT=RRANGE
ANGLE=ATAN(TOP/BOT)
C
DO 200 I=1,NMSK
WRITE (10,*) 'PUFF',I
IF (JSKIP.EQ.0) GO TO 300
C

```

```

C CYCLE THROUGH ALL DUST PUFFS
C TO EVALUATE DUST MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C BRACKETING EACH DUST PUFF WITHIN RANGE. DROP IF OUT OF ANGLE
C RANGE.
C THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE DUST PUFF.
C DROP PUFF IF NOT HIGH ENOUGH
C
C EVALUATE RANGE TO DUST PUFF
XP=DMSK(L1)
YP=DMSK(L2)
C
WRITE (10,*) 'XP YP',XP,YP
RADIUS=DMSK(L3)
C
WRITE (10,*) 'RADIUS',RADIUS
RP=((YP-YS)**2+(XP-XS)**2)**.5
C
WRITE (10,*) 'PUFF RANGE',RP, 'TARGET RANGE',RRANGE
IF ((RP-RADIUS).GT.RRANGE) GO TO 200
C
C EVALUATE ANGLES TO DUST PUFF
TOP=YP-YS
BOT=XP-XS
IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
IF (BOT.LT.0.0) THEN
IF (TOP.LT.0.0) DIR=DIR+180.
IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
IF (TOP.LT.0.0) DIR=360.-DIR
ENDIF
C
CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (DIR,AZIN)
IF (RADIUS.LT.0.030) RADIUS=0.030
IF (RP.LT.RADIUS) RP=RADIUS
DELAZ=ASIN(RADIUS/RP)*180./PI
AZLFT=AZIN-DELAZ
AZRGT=AZIN+DELAZ
IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.0
IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
C
WRITE (10,*) 'AZLEFT',AZLFT, 'AZRIGHT',AZRGT, 'AZIMUTH',AZIMUTH
IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200
C
C EVALUATE HEIGHT OF DUST PUFF
C
C FIND ELEVATION OF THE GROUND UNDER THIS PUFF
IXP=NINT(XP/SCALE)
IYP=NINT(YP/SCALE)
IF (IXP.LT.1) IXP=1
IF (IYP.LT.1) IYP=1
IF (IXP.GT.NX) IXP=NX
IF (IYP.GT.NY) IYP=NY
PX2=IXP*SCALE
PY2=IYP*SCALE
PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
IP=NY*(IXP-1)+IYP
PELEV=CONT(IELEV(IP))
C
C ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C MAX PUFF HEIGHT IS 30 METERS (100 FEET)
HEIGHT=RADIUS
IF (RADIUS.GT.0.030) HEIGHT=0.030
PELEV=PELEV+HEIGHT
C
C FIND ELEVATION ANGLE TO THIS DUST PUFF
TOP=PELEV-SELEV
BOT=RRANGE
BOT=RP
C
ANGLEI=ATAN(TOP/BOT)
WRITE (10,*) 'TARGET ELEVATION ANGLE',ANGLE
C
WRITE (10,*) 'PUFF ELEVATION ANGLE',ANGLEI
IF (ANGLEI.GT.ANGLE) JSKIP=0
200 CONTINUE
C
ENDIF
300 RETURN
END

SUBROUTINE ESMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C
C FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS WITHIN
C MAX RANGE OF THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
C
COMMON/MASKES/ESMSK(2400,3),NESMSK
COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
C
C XS,YS - SHOOTER COORDINATES
C RMAX1,2,3 - MAX RANGE OF SYSTEM WEAPONS
C ESMK(L1) - X COORD OF EACH ENGINE SMOKE PUFF
C ESMK(L2) - Y COORD OF EACH ENGINE SMOKE PUFF
C ESMK(L3) - RADIUS FOR EACH ENGINE SMOKE PUFF
C NESMSK - NUMBER OF ENGINE SMOKE PUFFS WITHIN MAX RANGE
C
PL=3.14159

```

```

NESMSK=0
IF (NENG.GT.0) THEN
C   CYCLE THROUGH ALL SMOKE PUFFS TO FILL MASKING ARRAY
RMAXX=0
IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
IF (RMAX3.GT.RMAXX) RMAXX=RMAX2
RMAXX=RMAXX/1000.
DO 20 I=ENG1,ENG2
  XD=PTENG(I,1)
  YD=PTENG(I,2)
  RADIUS=PTENG(I,3)*EXPUFF
  RANGE=((XD-XS)**2+(YD-YS)**2)**.5
  IF ((RANGE-RADIUS).GT.RMAXX) GO TO 20
  NESMSK=NESMSK+1
  FIND ANGLE TO THIS SMOKE PUFF
  TOP=YD-YS
  BOT=XD-XS
  IF (BOT.EQ.0.0) BOT=BOT+.001
  DIR=ATAN(ABS(TOP/ABS(BOT)))*180./PI
  IF (BOT.LT.0.0) THEN
    IF (TOP.LT.0.0) DIR=DIR+180.
    IF (TOP.GT.0.0) DIR=180.-DIR
  ELSE
    IF (TOP.LT.0.0) DIR=360.-DIR
  ENDF
  CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
  CALL FAZ (DIR,AZIN)
  ESMSK(NESMSK,1)=XD
  ESMSK(NESMSK,2)=YD
  ESMSK(NESMSK,3)=RADIUS
20 CONTINUE
ENDD
RETURN
END

SUBROUTINE ESSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
EVALUATES ENGINE SMOKE MASKING BETWEEN SHOOTER AND TARGET

XS,YS  SHOOTER REAL WORLD COORDINATES
XT,YT  TARGET REAL WORLD COORDINATES
AZIN  AZIMUTH TO TARGET
JSKIP  FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

COMMON/ELEVA/CONT(20),IELEV(10000)
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/MASKES/ESMSK(2400,3),NESMSK

DOUBLE PRECISION ANGLE,ANGLEI
PI=3.14159

IF (NESMSK.GT.0) THEN
  FIND SHOOTER AND TARGET GRID COORDINATES
  TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
  CENTER TO CENTER ABOUT THE SCALE.
  PRANGE=((YT-YS)**2+(XT-XS)**2)**.5
  IXS=NINT(XS/SCALE)
  IYS=NINT(YS/SCALE)
  IXT=NINT(XT/SCALE)
  IYT=NINT(YT/SCALE)
  IF (IXS.LT.1) IXS=1
  IF (IXT.LT.1) IXT=1
  IF (IYS.LT.1) IYS=1
  IF (IYT.LT.1) IYT=1
  IF (IXS.GT.NX) IXS=NX
  IF (IXT.GT.NX) IXT=NX
  IF (IYS.GT.NY) IYS=NY
  IF (IYT.GT.NY) IYT=NY
  X1=IXS*SCALE
  X2=IXT*SCALE
  Y1=IYS*SCALE
  Y2=IYT*SCALE
  RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5

  FIND SHOOTER AND TARGET ELEVATION
  IS=NY*(IXS-1)+IYS
  SELEV=CONT(IELEV(IS))
  IT=NY*(IXT-1)+IYT
  TELEV=CONT(IELEV(IT))

  ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
  FOR VEHICLE HEIGHT
  SELEV=SELEV+.002
  TELEV=TELEV+.002

  FIND ELEVATION ANGLE TO THIS TARGET
  TOP=TELEV-SELEV
  BOT=PRANGE
  ANGLE=ATAN(TOP/BOT)

DO 200 I=1,NESMSK
  IF (JSKIP.EQ.0) GO TO 300

  CYCLE THROUGH ALL ENGINE SMOKE PUFFS
  TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
  BRACKETING EACH PUFF WITHIN RANGE, DROP IF OUT OF ANGLE RANGE.
  THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE PUFF.
  DROP PUFF IF NOT HIGH ENOUGH

```

```

C   EVALUATE RANGE TO SMOKE PUFF
XP=ESMSK(I,1)
YP=ESMSK(I,2)
RADIUS=ESMSK(I,3)
RP=((YP-YS)**2+(XP-XS)**2)**.5
IF ((RP-RADIUS).GT.PRANGE) GO TO 200

C   EVALUATE ANGLES TO SMOKE PUFF
TOP=YP-YS
BOT=XP-XS
IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP/ABS(BOT)))*180./PI
IF (BOT.LT.0.0) THEN
  IF (TOP.LT.0.0) DIR=DIR+180.
  IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
  IF (TOP.LT.0.0) DIR=360.-DIR
ENDD

C   CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (DIR,AZIN)
IF (RADIUS.LT.0.030) RADIUS=0.030
IF (RP.LT.RADIUS) RP=RADIUS
DELAZ=ASIN(RADIUS/RP)*180./PI
AZLFT=AZIN-DELAZ
AZRGT=AZIN+DELAZ
IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.0
IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200

C   EVALUATE HEIGHT OF DUST PUFF

C   FIND ELEVATION OF THE GROUND UNDER THIS PUFF
IXP=NINT(XP/SCALE)
IYP=NINT(YP/SCALE)
IF (IXP.LT.1) IXP=1
IF (IYP.LT.1) IYP=1
IF (IXP.GT.NX) IXP=NX
IF (IYP.GT.NY) IYP=NY
PX2=IXP*SCALE
PY2=IYP*SCALE
PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
IP=NY*(IXP-1)+IYP
PELEV=CONT(IELEV(IP))

C   ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
MAX PUFF HEIGHT IS 30 METERS (100 FEET)
HEIGHT=RADIUS
IF (RADIUS.GT.0.030) HEIGHT=0.030
PELEV=PELEV+HEIGHT

C   FIND ELEVATION ANGLE TO THIS DUST PUFF
TOP=PELEV-SELEV
BOT=PRANGE
ANGLEI=ATAN(TOP/BOT)
IF (ANGLEI.GT.ANGLE) JSKIP=0
CONTINUE

ENDD

300 RETURN
END

SUBROUTINE PKLESS (KSIDE,IS,IT,RNG,BORE,TOF,PK)
REDUCES PK FOR WEAPONS WHICH CANNOT RANGE TO THE TARGET
DUE TO DUST OR SMOKE OBSCURANTS ON THE LINE OF SIGHT

COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1B,L2M,L3M,L1RM
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& ,BXYP(200,2),RXYP(200,2),ISPB(200),ISPOR(200)

DIMENSION SIG(40),AREA(40)
CHARACTER*90 L1B,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
DATA SIG /0.,1.,2.,3.,4.,5.,6.,7.,8.,9.,1.,1.,1.,2.,1.,3.,1.,4.,1.,5.,
& 1.,6.,1.,7.,1.,8.,1.,9.,2.,2.,1.,2.,2.,3.,2.,4.,2.,5.,2.,6.,2.,7.,2.,8.,2.,9.,3.,1.,
& 3.,2.,3.,3.,4.,3.,5.,3.,6.,3.,7.,3.,8.,3.,9.,
DATA AREA /5.,5398.,5793.,6179.,6554.,6915.,7258.,758.,7881.,
& .8159.,8413.,8643.,8849.,9032.,9192.,9332.,9452.,9554.,9641.,
& .9713.,9772.,9821.,9861.,9893.,9918.,9938.,9953.,9965.,9974.,
& .9981.,9987.,9990.,9993.,9995.,9997.,9998.,9998.,9999.,9999.,
& 1.000/

PI=3.14159
RANGE=RNG*1000.

C   WRITE (',') (SIG(I),I=1,40)
C   WRITE (',') (AREA(I),I=1,40)

C   WRITE (',') 'RANGE BORE TOF PKIN'
C   WRITE (',') RANGE,BORE,TOF,PK

C   IF (KSIDE.EQ.1) THEN
C     BLUE SHOOTER / RED TARGET
SS=REAL(ISPB(IS))/3600.*1000.

```

```

      ST=REAL(1SPDR(IT)/3600.*1000.
CALL FDIR (REAL(1BSA(1S,2),AZS)
CALL FDIR (REAL(1RSA(IT,2),AZT)
AZS=AZS/180.*PI
AZT=AZT/180.*PI
XS=BXVA(1S,1)
YS=BXVA(1S,2)
XT=RXVA(IT,1)
YT=RXVA(IT,2)
ELSE
      SS=REAL(1SPDR(1S)/3600.*1000.
      ST=REAL(1SPDR(IT)/3600.*1000.
CALL FDIR (REAL(1BSA(1S,2),AZS)
CALL FDIR (REAL(1RSA(IT,2),AZT)
AZS=AZS/180.*PI
AZT=AZT/180.*PI
XS=RXVA(1S,1)
YS=RXVA(1S,2)
XT=BXVA(IT,1)
YT=BXVA(IT,2)
ENDIF
C  DEFINE UNIT VECTOR CONNECTING SHOOTER AND TARGET
RMAG=((XS-XT)**2+(YS-YT)**2)**.5
DX=XS-XT
DY=YS-YT
UI=DX/RMAG
UJ=DY/RMAG
C  FIND RELATIVE CROSSING SPEED OF SHOOTER AND TARGET
BY TAKING CROSS PRODUCT OF EACH VEHICLE VELOCITY VECTOR
ONTO THIS UNIT VECTOR. THEN TAKE DIFFERENCE IN CROSS
PRODUCTS.
C  WRITE (',') 'SS ST AZS AZT'
C  WRITE (',') 'SS,ST,AZS,AZT'
C  WRITE (',') 'UI UJ'
C  WRITE (',') 'UI,UJ'
AS=SS*COS(AZS)
BS=SS*SIN(AZS)
AT=ST*COS(AZT)
BT=ST*SIN(AZT)
C  THE CROSS PRODUCT OF THESE VECTOR QUANTITIES IS THE CROSSING
C  SPEED
SCROSS=ABS((AS*UJ-BT*UI)-(AT*UJ-BT*UI))
C  WRITE (',') 'SCROSS',SCROSS
C  FIND TRUE ANGULAR RATE OF TARGET
OMEGAT=SCROSS/RANGE
C  FIND TIME TO TARGET RANGE
TB=TOF
TT=RANGE/BORE*TOF
C  SINCE THERMAL SIGHTS HAVE A RETICLE, GUNNERS CAN USUALLY
ESTIMATE TARGET RANGE BASED ON THE SIZE OF THE TARGET WITHIN
THE RETICLE AND THEN BRACKET THE RANGE. THIS WAY A BETTER RANGE
ESTIMATION CAN BE ENTERED INTO THE FIRE CONTROL COMPUTER OTHER
THAN THE BASIC BATTLESIGHT RANGE. FOR THIS REASON, THE ENTERED
BATTLESIGHT RANGE IS ONLY USED TO GET TIME OF FLIGHT DATA. AT
THIS POINT, THE GUNNER ESTIMATES A BETTER BATTLESIGHT RANGE
BASED ON THE FOLLOWING RANGE BRACKETS.
TB=1200./BORE*TOF
IF (RANGE.GE.3500.) BORE=4000.
IF (RANGE.GE.2750.AND.RANGE.LT.3500.) BORE=3000.
IF (RANGE.GE.2250.AND.RANGE.LT.2750.) BORE=2500.
IF (RANGE.GE.1750.AND.RANGE.LT.2250.) BORE=2000.
IF (RANGE.GE.1350.AND.RANGE.LT.1750.) BORE=1500.
IF (RANGE.LT.1350.) BORE=1200.
TB=BORE/1200.*TB
C  FIND DISTANCE TARGET HAS CROSSED IN THIS TIME
DT=TT*SCROSS
C  FIND TRUE LEAD ANGLE TO TARGET (MILS)
AT=DT/RANGE*1000.
AT=SCROSS/BORE*TB
C  APPLY THE TRUE ANGULAR RATE TO THE BATTLESIGHT RANGE
TO FIND THE DISTANCE THE FIRE CONTROL COMPENSATES.
DB=RANGE*OMEGAT*TB
C  FIND LEAD ANGLE TO BATTLESIGHT TARGET (MILS)
AB=DB/BORE*1000.
AB=SCROSS/RANGE*TB
C  FIND THE DIFFERENCE IN THE LEAD ANGLES (MILS)
DELAIM=ABS(AT-AB)
C  WRITE (',') 'DELAIM',DELAIM
C  FIND THE AIMPOINT OFFSET DISTANCE (METERS)
DIST=DELAIM*RANGE/1000.
DIST=DELAIM*RANGE
C  WRITE (',') 'DIST',DIST
C  ASSUME WEAPON PKH=1.0 AND THEREFORE HIT PROBABILITY IS

```

```

C  EQUAL TO PK/S AS ENTERED ABOVE
C  FIND DEFLECTION AND RANGE ERROR STANDARD DEVIATIONS
DAREA=PK*.5
C  RAREA=DAREA
DO 10 I=1,39
PROB1=(AREA(I)-0.5)*2.0
PROB2=(AREA(I+1)-0.5)*2.0
IF (DAREA.LE.PROB2.AND.DAREA.GE.PROB1) GO TO 20
10 CONTINUE
20 CONTINUE
BOT=PROB2-PROB1
TOP=SIG(I+1)-SIG(I)
DSIG=SIG(I)+(DAREA-PROB1)*TOP/BOT
RSIG=DSIG
C  FIND TARGET DEFLECTION LEFT AND RIGHT LIMITS IN TERMS
OF STANDARD DEVIATIONS WHEN HORIZONTAL AIMPOINT ERROR IS INCLUDED
C  ASSUME TARGET IS THREE BY THREE METERS SQUARE
C  THEREFORE FIND ADDITIONAL DEFLECTION ERROR IN TERMS OF A TARGET
C  WIDTH STANDARD DEVIATION
DIST=ABS(DIST/1.5)
BLFT=DSIG-DIST
BRGT=DSIG+DIST
SLFT=ABS(BLFT)
SRGT=ABS(BRGT)
J=1
K=1
C  FIND PROBABILITIES ASSOCIATED WITH THESE SIGMA BOUNDARIES
DO 30 I=1,39
IF (SLFT.LE.SIG(I+1).AND.SLFT.GE.SIG(I)) THEN
A2=AREA(I)-0.5
J=I
ENDIF
IF (SRGT.LE.SIG(I+1).AND.SRGT.GE.SIG(I)) THEN
A1=AREA(I)-0.5
K=I
ENDIF
30 CONTINUE
TOP=AREA(J+1)-AREA(J)
BOT=SIG(J+1)-SIG(J)
A2=A2+(SLFT-SIG(J))*TOP/BOT
TOP=AREA(K+1)-AREA(K)
BOT=SIG(K+1)-SIG(K)
A1=A1+(SRGT-SIG(K))*TOP/BOT
IF (SLFT.GE.SIG(40)) A2=0.5
IF (SRGT.GE.SIG(40)) A1=0.5
IF (BRGT.GE.0.0) THEN
DPROB=A1+A2
ELSE
DPROB=A2-A1
ENDIF
C  VERTICAL AIMPOINT ERROR BASED ON TIME OF FLIGHT TO TRUE
C  RANGE WHEN AIMING AT BATTLESIGHT RANGE
G=9.807
C  FIND MUZZLE VELOCITY (M/S)
V0=BORE/TOF
V0=BORE/TB
C  FIND SUPER ELEVATION ANGLE TO TARGET CENTER AT BATTLESIGHT
C  RANGE
SE=0.5*ASIN(BORE/V0**2*G)
SE=1-SE*180./PI
C  WRITE (',') 'SUPER ELEVATION',SE1
C  FIND HEIGHT ABOVE OR BELOW TARGET CENTER AT TRUE TARGET RANGE
HEIGHT=G/2.*TT**2+V0*SIN(SE)*TT
C  WRITE (',') 'HEIGHT',HEIGHT
C  EVALUATE NEW RANGE ERROR PROBABILITIES
C  FIND TARGET RANGE TOP AND BOTTOM LIMITS IN TERMS
OF STANDARD DEVIATIONS WHEN THE VERTICAL AIMPOINT ERROR IS INCLUDED
C  ASSUME TARGET IS THREE BY THREE METERS SQUARE
VERR=ABS(HEIGHT/1.5)
BTOP=RSIG+VERR
BBOT=RSIG-VERR
C  FIND THESE BOUNDARIES IN TERMS OF STANDARD DEVIATIONS
STOP=ABS(BTOP)
SBOT=ABS(BBOT)
J=1
K=1
C  FIND PROBABILITIES ASSOCIATED WITH THESE SIGMA BOUNDARIES
DO 40 I=1,39
IF (STOP.LE.SIG(I+1).AND.STOP.GE.SIG(I)) THEN
A1=AREA(I)-0.5
J=I
ENDIF
IF (SBOT.LE.SIG(I+1).AND.SBOT.GE.SIG(I)) THEN
A2=AREA(I)-0.5
K=I

```

```

40  ENDF
    CONTINUE

    TOP=AREA(J+1)-AREA(J)
    BOT=SIG(J+1)-SIG(J)
    A1=A1+(STOP-SIG(J))*TOP/BOT

    TOP=AREA(K+1)-AREA(K)
    BOT=SIG(K+1)-SIG(K)
    A2=A2+(SBOT-SIG(K))*TOP/BOT

    IF (STOP.GE.SIG(40)) A1=0.5
    IF (SBOT.GE.SIG(40)) A2=0.5

    IF (STOP.GE.0.0) THEN
        RPROB=A1+A2
    ELSE
        RPROB=A2-A1
    ENDF

C   CALCULATE NEW PK BASED ON BATTLESIGHT RANGE ERROR
    PK=DPROB*RPROB

C   WRITE (*,*) 'PKOUT 'PK, ' DPROB 'DPROB, ' RPROB 'RPROB

    RETURN
    END

SUBROUTINE LIVING
C   FOR AIDING IN THE DECISION PROCESS WHETHER ONE WOULD
C   SHOOT OR HOLD FIRE EVEN WITH A VERY LOW PK, HAVING A
C   ROUGH IDEA OF HOW MANY THREAT TARGETS EXIST COMPARED TO
C   HOW MANY FRIENDLY VEHICLES EXIST IS A REASONABLE FACTOR
C   TO CONSIDER

    COMMON/ALIVE/NBA,NRA,BTHREAT,RTHREAT
    COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
    & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,L2RF,L3RF,NB,NR

    CHARACTER*80 L1BF,L2F,L3F,L1RF,L2RF,L3RF,L1RM,L2RM,L3RM,L1RM
    CHARACTER*16 BHIER,RHIER,IHIER
    CHARACTER*24 BUNIT,RUNIT

    NBA=0
    NRA=0
    DO 10 I=1,NB
        IF (IBSYS(I,2).EQ.1.AND.IBSYS(I,1).LE.10) NBA=NBA+1
    DO 20 I=1,NR
        IF (IRSYS(I,2).EQ.1.AND.IRSYS(I,1).LE.10) NRA=NRA+1

    EVALUATE THREAT PICTURE BASED ON NUMBER OF THEM VERSUS NUMBER OF
    YOU. IF THREAT=1, ALWAYS SHOOT; IF THREAT=0, CONSIDER PK AND
    AMMUNITION FACTORS.

    BTHREAT=1
    RTHREAT=1

    RETURN
    END

SUBROUTINE GRMASK (XS,YS,RMAX1,RMAX2,RMAX3)
C   FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS WITHIN
C   MAX RANGE OF THIS SYSTEM SO THAT MASKING CAN BE EVALUATED

    COMMON/MASKGR/GRMSK(400,3),NGRMSK
    COMMON/PUFFS/TWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
    & IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
    & IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
    & TASMK(400,2),PART(96,2),TAART(96,2),NDUSTMX,NENGMX,NSMKMX,
    & NARTMX

C   XS,YS - SHOOTER COORDINATES
C   RMAX1,2,3 - MAX RANGE OF SYSTEM WEAPONS
C   GRMSK(I,1) - X COORD OF EACH CURTAIN CENTER
C   GRMSK(I,2) - Y COORD OF EACH CURTAIN CENTER
C   GRMSK(I,3) - ORIENTATION ANGLE (CARTESIAN) OF CURTAIN
C   NGRMSK - NUMBER OF SMOKE GRENADE CURTAINS WITHIN MAX RANGE

    PI=3.14159
    NGRMSK=0
    IF (NSMK.GT.0) THEN
        CYCLE THROUGH ALL SMOKE PUFFS TO FILL MASKING ARRAY
        RMAXX=0
        IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
        IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
        IF (RMAX3.GT.RMAXX) RMAXX=RMAX3
        RMAXX=RMAXX/1000.
        DO 20 I=ISMK1,ISMK2
            XD=PSMK(I,1)
            YD=PSMK(I,2)
            RANGE=((XD-XS)**2+(YD-YS)**2)**.5
            IF (RANGE.GT.RMAXX) GO TO 20
            NGRMSK=NGRMSK+1
            CALL FDIR (TASMK(I,2),GRMSK(I,3))

```

```

20  GRMSK(NGRMSK,1)=XD
    GRMSK(NGRMSK,2)=YD
    CONTINUE
    ENDF
    RETURN
    END

SUBROUTINE GRSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
EVALUATES ENGINE SMOKE MASKING BETWEEN SHOOTER AND TARGET

C   XS,YS SHOOTER REAL WORLD COORDINATES
C   XT,YT TARGET REAL WORLD COORDINATES
C   AZIN AZIMUTH TO TARGET
C   JSKIP FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

    COMMON/ELEVA/CONT(20),IELEV(10000)
    COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
    COMMON/MASKGR/GRMSK(400,3),NGRMSK

    DOUBLE PRECISION ANGLE,ANGLEI

    PI=3.14159

    IF (NGRMSK.GT.0) THEN
        CALL FDIR (AZIMUTH,DIRT)
        WCURT=180.*3048/1000.

C   FIND SHOOTER AND TARGET GRID COORDINATES
C   TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C   CENTER TO CENTER ABOUT THE SCALE.
        RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
        IXS=NINT(XS/SCALE)
        IYS=NINT(YS/SCALE)
        IXT=NINT(XT/SCALE)
        IYT=NINT(YT/SCALE)
        IF (IXS.LT.1) IXS=1
        IF (IXT.LT.1) IXT=1
        IF (IYS.LT.1) IYS=1
        IF (IYT.LT.1) IYT=1
        IF (IXS.GT.NX) IXS=NX
        IF (IXT.GT.NX) IXT=NX
        IF (IYS.GT.NY) IYS=NY
        IF (IYT.GT.NY) IYT=NY
        X1=IXS*SCALE
        X2=IXT*SCALE
        Y1=IYS*SCALE
        Y2=IYT*SCALE
        RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5

C   FIND SHOOTER AND TARGET ELEVATION
        IS=NY*(IXS-1)+IYS
        SELEV=CONT(IELEV(IS))
        IT=NY*(IXT-1)+IYT
        TELEV=CONT(IELEV(IT))

C   ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
C   FOR VEHICLE HEIGHT
        SELEV=SELEV+.002
        TELEV=TELEV+.002

C   FIND ELEVATION ANGLE TO THIS TARGET
        TOP=TELEV-SELEV
        BOT=RRANGE
        ANGLE=ATAN(TOP/BOT)

C   DO 200 I=1,NGRMSK
        IF (JSKIP.EQ.0) GO TO 300

C   CYCLE THROUGH ALL SMOKE GRENADE CURTAINS
C   TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
C   BRACKETING EACH CURTAIN WITHIN RANGE. DROP IF OUT OF ANGLE RANGE.
C   THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE CURTAIN.
C   DROP IF NOT HIGH ENOUGH
        EVALUATE RANGE TO CURTAIN
        XP=GRMSK(I,1)
        YP=GRMSK(I,2)
        RP=((YP-YS)**2+(XP-XS)**2)**.5
        IF (RP.GT.RRANGE) GO TO 200

C   EVALUATE ANGLES TO SMOKE CURTAIN
        TOP=YP-YS
        BOT=XP-XS
        IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
        IF (BOT.EQ.0.0) BOT=BOT+.001
        DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
        IF (BOT.LT.0.0) THEN
            IF (TOP.LT.0.0) DIR=DIR+180.
            IF (TOP.GT.0.0) DIR=180.-DIR
        ELSE
            IF (TOP.LT.0.0) DIR=360.-DIR
        ENDF

C   FIND LEFT AND RIGHT LIMIT ANGLES OF THIS CURTAIN.
C   FIND THE INCLUDED ANGLE BETWEEN THE CURTAIN AND THE RANGE
C   TO THE CURTAIN.
C   TO DO THIS, TAKE THE CROSS PRODUCT OF THE CURTAIN VECTOR
C   ON TO THE UNIT VECTOR DEFINING THE RANGE TO THE CURTAIN.
        RI=BOT/RP*CCS(DIR/180.*PI)
        RJ=TOP/RP*SIN(DIR/180.*PI)
        CI=WCURT/2.*CCS(GRMSK(I,3)/180.*PI)

```



```

C
GO TO 141
1000 CONTINUE
RETURN
END

SUBROUTINE GRADE (KSIDE,K)
COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
COMMON/ELEV/CONT(20),IELEV(10000)
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1B,L2M,L3M,L1RM
CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
P=3.14159

IF (KSIDE.EQ.1) THEN
  XS=BXYA(K,1)
  YS=BXYA(K,2)
  AZ=BSA(K,2)
ELSE
  XS=RXYA(K,1)
  YS=RXYA(K,2)
  AZ=RSA(K,2)
ENDIF

C CONVERT DIRECTION AZIMUTH TO CARTESIAN ANGLE
CALL FOUR (AZ,DIR)

C FIND VEHICLE GRID COORDINATES
TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
C CENTER TO CENTER ABOUT THE SCALE.
IXS=NINT(XS/SCALE)
IYS=NINT(YS/SCALE)
IF (IXS.LT.1) IXS=1
IF (IYS.LT.1) IYS=1
IF (IXS.GT.NX) IXS=NX
IF (IYS.GT.NY) IYS=NY
X1=IXS*SCALE
Y1=IYS*SCALE

C FIND NEXT GRID OVER ON THE DIRECTION OF MOVEMENT
XT=XS+SCALE*COS(DIR/180.*PI)
YT=YS+SCALE*SIN(DIR/180.*PI)
IXT=NINT(XT/SCALE)
IYT=NINT(YT/SCALE)
IF (IXT.LT.1) IXT=1
IF (IYT.LT.1) IYT=1
IF (IXT.GT.NX) IXT=NX
IF (IYT.GT.NY) IYT=NY
X2=IXT*SCALE
Y2=IYT*SCALE

RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5

C FIND VEHICLE AND NEXT GRID ELEVATION (IT'S ALREADY IN KM)
IS=NY*(IXS-1)+IYS
SELEV=CONT(IELEV(IS))
IT=NY*(IXT-1)+IYT
TELEV=CONT(IELEV(IT))

C TOP=TELEV-SELEV
BOT=RANGE
IF (TOP.EQ.0.) GRAD=0.0
IF (BOT.EQ.0.) GRAD=0.0
IF (TOP.NE.0.0.AND.BOT.NE.0.0) GRAD=ATAN(TOP/BOT)*180/PI

IF (KSIDE.EQ.1) THEN
  GRADEB(K)=GRAD
ELSE
  GRADER(K)=GRAD
ENDIF

RETURN
END

SUBROUTINE FINDACEL (KSIDE,K,ARATE)
COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1B,L2M,L3M,L1RM
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& BXYP(200,2),RXPY(200,2),ISPB(200),ISPR(200)
COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT

IDT=IDTIME

IF (KSIDE.EQ.1) THEN
  GRADE=GRADEB(K)
  ISWANT=BSA(K,1)
  ISNOW=ISPB(K)
  NSYS=IRSYS(K,1)
  IF (NSYS.GT.10) NSYS=1
  VO=ACELB(NSYS,1)
  VG=ACELB(NSYS,3)
  AO=VO/ACELB(NSYS,2)
  AG=VG/ACELB(NSYS,5)
  G=ACELB(NSYS,4)
ELSE
  GRADE=GRADER(K)
  ISWANT=IRSA(K,1)
  ISNOW=ISPR(K)
  NSYS=IRSYS(K,1)
  IF (NSYS.GT.10) NSYS=1
  VO=ACELR(NSYS,1)
  VG=ACELR(NSYS,3)
  AO=VO/ACELR(NSYS,2)
  AG=VG/ACELR(NSYS,5)
  G=ACELR(NSYS,4)
ENDIF

C WRITE ('.') 'GRADE ISWANT ISNOW VO VG AO AG'
C WRITE ('.') 'GRADE,ISWANT,ISNOW,VO,VG,AO,AG'

C CHANGE IN ACCELERATION WITH RESPECT TO GRADE
DADG=(AG-AO)/G

C MAX ALLOWABLE ACCELERATION BASED ON CURRENT GRADE
AMAX=DADG*GRADE+AO
IF (AMAX.LT.0.0) AMAX=0.0
IF (AMAX.GT.AO) AMAX=AO

C MAX ALLOWABLE SPEED BASED ON CURRENT GRADE
VMAX=(VG-VO)/G*GRADE+VO
IF (VMAX.LT.0.0) VMAX=0.0
IF (VMAX.GT.VO) VMAX=VO
IVMAX=INT(VMAX)

C IF DESIRED SPEED IS GREATER THAN ALLOWABLE SPEED, TRUNCATE
IF (ISWANT.GT.IVMAX) ISWANT=IVMAX

C FIND DIFFERENCE BETWEEN ACTUAL AND DESIRED SPEED
IDV=ISWANT-ISNOW

C CONVERT TO AN ACCELERATION WITHIN THIS TIME STEP
DVDI=REAL(IDV/IDT)

ARATE=DVDI
IF (ARATE.GT.AMAX) ARATE=AMAX

C WRITE ('.') 'GRADE ISWANT ISNOW IVMAX AMAX ARATE'
C WRITE ('.') 'GRADE,ISWANT,ISNOW,IVMAX,AMAX,ARATE'

RETURN
END

SUBROUTINE STATUS
C
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1B,L2M,L3M,L1RM
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRNB(30),NRNR(30),TEMAB(30),TEMAR(30),TEMNB(30),TEMNR(30),
& TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MILB(10,3),MILR(10,3)
COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& BXYP(200,2),RXPY(200,2),ISPB(200),ISPR(200)
COMMON/FIRE/IBPRIME(200,2),IBSECOND(200,2),IRPRIME(200,2),
& IRSECOND(200,2),WRNGB(200,2),WRNGR(200,2)
COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMNR(200,3),TTMMB(200,3),TTMNR(200,3),MODEFB(200),
& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,3),TCFMR(200,3),
& TTFMB(200,3),TTFMR(200,3),TRMMB(200),TRMMR(200),
& TRMB(200),TROESB(200),TROESR(200)
COMMON/BATTLE/ITIME,IDTIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
COMMON/ALIVE/NBA,NRA,BTHREAT,RTHREAT
COMMON/ALIVE2/IBA(10),IRA(10),IBT(10),IRT(10),DELB(10),DELR(10)
COMMON/ENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
& ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)

C CHARACTER*8 BFNAME,RFNAME
C CHARACTER*80 LINE
C CHARACTER*16 BHIER,RHIER,IHIER
C CHARACTER*24 BUNIT,RUNIT
C CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C CHARACTER*1 C2
C CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK
C
C WRITE ('.')
C KSIDE=0
C WRITE ('.') 'CURRENT STATUS OF FORCES'
C WRITE ('.') 'enter BLUE or RED, CUMULATIVE or QUIT'
C WRITE ('.')
C READ ('.',ERR=10) C2

```







```

C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C DO 800 I=1,10
C READ (1,303) LINE
C WRITE (*,503) LINE
C DO 800 I=1,3
C IF (I-1) 3=1
C IF (IC.EQ.1) READ (1,321) (BWSIG(I,J),J=1,10)
C IF (IC.EQ.1) WRITE (*,521) (BWSIG(I,J),J=1,10)
C IF (IC.EQ.2) READ (1,321) (RWSIG(I,J),J=1,10)
C IF (IC.EQ.2) WRITE (*,521) (RWSIG(I,J),J=1,10)
C 800 CONTINUE
C 900 CONTINUE
C 1000 CONTINUE
C CLOSE (1)
C RETURN
C END

```

#### SUBROUTINE CHECKSIG (KSIDE,K,J,R,ISIG)

CHECKS TO SEE IF TARGET VEHICLE IS DETECTABLE BASED ON SIGNATURES

IF ISIG IS RETURNED AS 0 THEN TARGET SIGNATURE IS NOT DETECTED, ELSE ISIG = 1

```

KSIDE SEARCHER SIDE
K SEARCHER NUMBER
J TARGET NUMBER
R TARGET RANGE (KM)
ISIG SIGNATURE FLAG

```

```

COMMON/SIGTURS/RSMIN,RSMAX,BDETCT(10,10),RDETCT(10,10),
& BWSIG(30,10),RWSIG(30,10)
COMMON/MOVE/XYA(200,2),RXYA(200,2),ISA(200,2),RSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/MANEUVER/XYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& BXYP(200,2),RXYP(200,2),ISPD8(200),ISPD8(200)
COMMON/OSCORE/NGREN8(200,2),NGREN8(200,2),IBESMK(200),
& IBESMK(200)
COMMON/MODES/MODEM8(200),MODEMR(200),TCMM8(200,3),
& TCMMR(200,3),TTMM8(200,3),TTMMR(200,3),MODEFB(200),
& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMFB(200),TRMMR(200),
& TRMFR(200),TRDES8(200),TRDES8(200)
COMMON/BATTLE/ITIME,ITIME,ITGTR(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,L2RF,L3RF,
COMMON/DEFILADE/IDEFB(200),IDEFR(200)

```

```

CHARACTER*30 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
CHARACTER*18 BHIER,RHIER,INIER
CHARACTER*24 BUNIT,RUNIT

```

ISIG=0

RANGE=R\*1000.0

```

C SEE IF TARGET IS MOVING
C IF (KSIDE.EQ.1) ISPD=ISPD8(J)
C IF (KSIDE.EQ.2) ISPD=ISPD8(J)
C IF (ISPD.EQ.1) ISIG=1
C IF (ISIG.EQ.1) RETURN

C SEE IF TARGET IS BLOWING ENGINE SMOKE OR FIRING SMOKE GRENADES
C IF (KSIDE.EQ.1) THEN
C IBLOW1=IBESMK(J)
C IBLOW2=NGREN8(J,2)
C ELSE
C IBLOW1=IBESMK(J)
C IBLOW2=NGREN8(J,2)
C ENDIF
C IF (IBLOW1.EQ.1.OR.IBLOW2.EQ.1) ISIG=1
C IF (ISIG.EQ.1) RETURN

C SEE IF TARGET IS FIRING A WEAPON AND IF SIGNATURE IS DETECTED
C IF (KSIDE.EQ.1) MODEF=MODEFR(J)
C IF (KSIDE.EQ.2) MODEF=MODEFB(J)
C IF (MODEF.EQ.2.OR.MODEF.EQ.8.OR.MODEF.EQ.8) THEN
C TARGET HAS ACTIVE WEAPONS
C FIND WEAPON NUMBER AND ITS SIGNATURE
C IF (KSIDE.EQ.1) THEN
C ITWP=ITGTR(J,2)
C ITSYS=IRSYS(J,1)
C MYSYS=IBSYS(K,1)
C IWANT=(ITSYS-1)*9+ITWP
C DFACT=RWSIG(IWANT,MYSYS)
C ELSE
C ITWP=ITGTR(J,2)
C ITSYS=IBSYS(J,1)
C MYSYS=IRSYS(K,1)
C IWANT=(ITSYS-1)*9+ITWP
C DFACT=BWSIG(IWANT,MYSYS)
C ENDIF

```

#### INTERPOLATE BASED ON RANGE

```

TOP=DFACT-1.0
BOT=RSMAX-RSMIN
SLOPE=TOP/BOT
B=1.0-SLOPE*RSMIN
DFACT=B+SLOPE*RANGE
IF (DFACT.LT.0.0) DFACT=0.0
IF (DFACT.GT.1.0) DFACT=1.0

```

#### EVALUATE THIS DETECTION PROBABILITY

```

C DRAND=RND()
C WRITE (10,7) LAUNCH SIGNATURE: DFACT PROB
C WRITE (10,7) DFACT,DRAND
C IF (DFACT.GT.DRAND) ISIG=1
C ENDIF
C IF (ISIG.EQ.1) RETURN

```

#### SINCE TARGET IS NOT MOVING OR MAKING ITSELF OBVIOUS

EVALUATE BASIC TARGET DETECTION PROBABILITY

RDEF=1.

IF (KSIDE.EQ.1) THEN

ITSYS=IRSYS(J,1)

MYSYS=IBSYS(K,1)

DFACT=BDETCT(MYSYS,ITSYS)

IF TARGET IS IN DEFILADE THIS DETECTION FACTOR WILL BE HALVED

IF (IDEFR(J).EQ.1) RDEF=2.

ELSE

ITSYS=IBSYS(J,1)

MYSYS=IRSYS(K,1)

DFACT=BDETCT(MYSYS,ITSYS)

IF (IDEFB(J).EQ.1) RDEF=2.

ENDIF

#### INTERPOLATE BASED ON RANGE

TOP=DFACT-1.0

BOT=RSMAX-RSMIN

SLOPE=TOP/BOT

B=1.0-SLOPE\*RSMIN

DFACT=B+SLOPE\*RANGE

DFACT=DFACT/RDEF

IF (DFACT.LT.0.0) DFACT=0.0

IF (DFACT.GT.1.0) DFACT=1.0

#### EVALUATE THIS DETECTION PROBABILITY

C DRAND=RND()

C WRITE (10,7) DETECTION SIGNATURE: DFACT PROB

C WRITE (10,7) DFACT,DRAND

C IF (DFACT.GT.DRAND) ISIG=1

RETURN

END

#### SUBROUTINE READAPS

READS ACTIVE PROTECTION SYSTEM FILE INFORMATION

COMMON/APS/IBSAPS(10),IRSAPS(10),BAPSF(10,3),RAPSF(10,3),

& BAPST,RAPST,IRAMS8(2),IRAMSR(2),IAPSON8(200,2),IAPSONR(200,2)

#### CHARACTER\*30 LINE

```

303 FORMAT (A30)
503 FORMAT (1X,A30)
OPEN (1,FILE='APS.IN')
READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
313 FORMAT (10I8)
314 FORMAT (1X,10I8)
C READ (1,313) (IBSAPS(K),K=1,10)
C WRITE (*,314) (IBSAPS(K),K=1,10)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,313) (IRSAPS(K),K=1,10)
C WRITE (*,314) (IRSAPS(K),K=1,10)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
320 FORMAT (10F8,2)
520 FORMAT (1X,10F8,2)
C READ (1,320) (BAPSF(K,1),K=1,10)
C READ (1,320) (BAPSF(K,2),K=1,10)
C READ (1,320) (BAPSF(K,3),K=1,10)
C WRITE (*,520) (BAPSF(K,1),K=1,10)
C WRITE (*,520) (BAPSF(K,2),K=1,10)
C WRITE (*,520) (BAPSF(K,3),K=1,10)
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,303) LINE
C WRITE (*,503) LINE
C READ (1,320) (RAPSF(K,1),K=1,10)
C READ (1,320) (RAPSF(K,2),K=1,10)
C READ (1,320) (RAPSF(K,3),K=1,10)
C WRITE (*,520) (RAPSF(K,1),K=1,10)

```

```

C WRITE (I,820) (RAPS(K,2),K=1,10)
C WRITE (I,820) (RAPS(K,3),K=1,10)
C READ (1,303) LINE
C WRITE (I,803) LINE
C READ (1,303) LINE
C WRITE (I,803) LINE
C READ (1,303) LINE
C WRITE (I,803) LINE
C 830 FORMAT (F5.2,2H)
C 831 FORMAT (1X,F5.2,2H)
C READ (1,830) BAPST,(IRAMSB(K),K=1,2)
C WRITE (I,831) BAPST,(IRAMSB(K),K=1,2)
C READ (1,303) LINE
C WRITE (I,803) LINE
C READ (1,303) LINE
C WRITE (I,803) LINE
C READ (1,303) LINE
C WRITE (I,803) LINE
C READ (1,830) RAPST,(IRAMSR(K),K=1,2)
C WRITE (I,831) RAPST,(IRAMSR(K),K=1,2)
C CLOSE (I)
C INITIALIZE ACTIVE PROTECTION TO OFF FOR ALL VEHICLES
DO 800 I=1,200
  IAPSONB(I,2)=IRAMSB(2)
  IAPSONR(I,2)=IRAMSR(2)
  IAPSONB(I,1)=0
  IAPSONR(I,1)=0
800 RETURN
END

SUBROUTINE CHECKAPS (KSIDE,KSYS,J,KWP,PK,IFAPS,RNG)
CHECKS TO SEE IF TARGET VEHICLE HAS ACTIVE PROTECTION
TURNED ON AND CAN COUNTER THIS ATTACK

ATTACKER PK IS SET TO 0 IF ACTIVE PROTECTION DEFEATS THIS
ATTACK

KSIDE SHOOTER SIDE
KSYS SHOOTER SYSTEM NUMBER
J TARGET NUMBER
KWP SHOOTER WEAPON NUMBER
PK SHOOTER PK AGAINST THIS TARGET
IFAPS FLAG FOR ACTIVE PROTECTION (SEE OUTPUT1)
ACTIVE PROTECTION FLAG (1 INTERCEPT, 0 MISS, -1
N/A THIS WEAPON, 2 OUT OF SHOTS, 3 NO TIME)

COMMON/APS/BSAPS(10),IRSAPS(10),BAPSF(10,3),RAPSF(10,3),
& BAPST,RAPST,IRAMSB(2),IRAMSR(2),IAPSONB(200,2),IAPSONR(200,2)
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& BSYS(200,2),RSYS(200,2),L1BF,L2F,L3F,L1RF,L2RF,L3RF,
& COMMON/SIMAPS/ISMB(200),ISMR(200)
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRNB(30),NRNR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
& TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MILB(10,3),MILR(10,3)

CHARACTER*8 BFNAME,RFNAME
CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT

IF (KSIDE.EQ.1) THEN
  ITSYS=IRSYS(J,1)
  HVAPS=IRSAPS(ITSYS)
  FAPS=RAPSF(KSYS,KWP)
  NSHOTS=IAPSONR(J,2)
  LIVE=IRSYS(J,2)
  ISM=ISMR(J)
  ISMAX=IRAMSR(1)
ELSE
  ITSYS=BSYS(J,1)
  HVAPS=BSAPS(ITSYS)
  FAPS=BAPSF(KSYS,KWP)
  NSHOTS=IAPSONB(J,2)
  LIVE=BSYS(J,2)
  ISM=ISMB(J)
  ISMAX=IRAMSB(1)
ENDIF

C ACTIVATE ACTIVE PROTECTION BASED ON SIMULTANEITY AND TIME TO
C IMPACT

IAMON=0
TRNG=RNG*1000.

IF (ISM.LT.ISMAX) THEN

C FIND TIME OF FLIGHT TO SEE IF APS COULD REACT BEFORE HIT
IF (KSIDE.EQ.1) THEN
  IST=(KSYS-1)*3-KWP
  NRQ=NRWB(KSYS,KWP)
  ISRM=(KSYS-1)*18+(KWP-1)*6+NRQ
  ISR1=(KSYS-1)*18+(KWP-1)*6+1
  RMIN=RB(ISR1)
  RMAX=RR(ISRM)
  TMIN=TEMINB(IST)
  TMAX=TEMAXB(IST)
  CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,ETIME)
  THIT=ETIME-TMIN

```

```

ELSE
  IST=(KSYS-1)*3-KWP
  NRQ=NRWR(KSYS,KWP)
  ISRM=(KSYS-1)*18+(KWP-1)*6+NRQ
  ISR1=(KSYS-1)*18+(KWP-1)*6+1
  RMIN=RR(ISR1)
  RMAX=RR(ISRM)
  TMIN=TEMINR(IST)
  TMAX=TEMAXR(IST)
  CALL TSHOOT (RMIN,RMAX,TMIN,TMAX,TRNG,ETIME)
  THIT=ETIME-TMIN
ENDIF

IF (KSIDE.EQ.1) THEN
  JSYS=IRSYS(J,1)
  FAPS=RAPSF(KSYS,KWP)
  IF (IRSAPS(JSYS).EQ.1) THEN
    TMIN=RAPST
    IF (THIT.GE.TMIN.AND.FAPS.GT.0.0) THEN
      ISMR(J)=ISMR(J)+1
      IAMON=1
    ENDIF
  ELSE
    JSYS=BSYS(J,1)
    FAPS=BAPSF(KSYS,KWP)
    IF (BSAPS(JSYS).EQ.1) THEN
      TMIN=BAPST
      IF (THIT.GE.TMIN.AND.FAPS.GT.0.0) THEN
        ISMB(J)=ISMB(J)+1
        IAMON=1
      ENDIF
    ENDIF
  ENDIF
ENDIF
ENDIF

```

```

C THE ORDER TO THE FOLLOWING LOGIC IS CRITICAL TO SELECTING
C THE CORRECT FLAG

```

```

IF (NSHOTS.EQ.0) IFAPS=2
IF (IAMON.EQ.0) IFAPS=3
IF (HVAPS.EQ.0) IFAPS=1
IF (FAPS.LE.0) IFAPS=-1

```

```

C ACTIVE PROTECTION FLAG (1 INTERCEPT, 0 MISS, -1
C N/A THIS WEAPON, 2 OUT OF SHOTS, 3 NO TIME)

```

```

IF (HVAPS.EQ.1.AND.IAMON.EQ.1.AND.NSHOTS.GE.1.AND.LIVE.EQ.1) THEN
  APS CAN RESPOND
  PROB=RND0
  C WRITE (10,*) 'APS PROB'
  C WRITE (10,*) FAPS,PROB
  IF (FAPS.GT.PROB) THEN
    PK=0.0
    IFAPS=1
    IF (KSIDE.EQ.1) THEN
      WRITE (10,*) 'RED 'J,' FIRES APS - INTERCEPT '
    ELSE
      WRITE (10,*) 'BLUE 'J,' FIRES APS - INTERCEPT '
    ENDIF
  ELSE
    IFAPS=0
    IF (KSIDE.EQ.1) THEN
      WRITE (10,*) 'RED 'J,' FIRES APS - MISS ....'
    ELSE
      WRITE (10,*) 'BLUE 'J,' FIRES APS - MISS ....'
    ENDIF
  ENDIF
  IF (KSIDE.EQ.1) THEN
    IAPSONR(J,2)=IAPSONR(J,2)-1
  ELSE
    IAPSONB(J,2)=IAPSONB(J,2)-1
  ENDIF
ENDIF
RETURN
END

```

```

SUBROUTINE OUTPUT1 (KTIME,KSIDE,K,KSYS,KWP,ENERGY,RANGE,J,JSYS,
& IFAPS,KILL)

```

```

C WRITE OUTPUT DATA FOR POST PROCESSING ON SHOTS TAKEN DURING
C BATTLE

```

```

C OUTPUT 1:

```

```

C KTIME KILL TIME
C KSIDE SHOOTER SIDE (1 BLUE, 2 RED)
C K SHOOTER NUMBER
C KSYS SHOOTER SYSTEM
C KWP SHOOTER WEAPON
C ENERGY BLUE EM GUN ENERGY % SHOT USED
C RANGE TARGET RANGE
C J TARGET NUMBER
C JSYS TARGET SYSTEM
C AZI ATTACK AZIMUTH ON TARGET TURRET
C AZH ATTACK AZIMUTH ON TARGET HULL
C IFAPS ACTIVE PROTECTION FLAG (1 INTERCEPT, 0 MISS, -1
C N/A THIS WEAPON, 2 OUT OF SHOTS, 3 NO TIME)
C KILL TARGET KILLED FLAG (1 - KILLED, 0 - MISSED)
C AZOUT SHOT AZIMUTH RELATIVE TO SHOOTER HULL FRONT

```

```

COMMON/IRE/ISPRIME(200,2),ISSECOND(200,2),IRPRIME(200,2),
& IRSECOND(200,2),WRNGS(200,2),WRNGR(200,2)
COMMON/MOVE/BXYA(200,2),RXYA(200,2),ISBA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
& TRMFR(200),TRDES(200),TRDESR(200)
COMMON/BATTLE/ITIME,ITIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
COMMON/TURRET/AZTB(200),AZTR(200),TRATEB(10),TRATER(10)

CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C FIND ATTACK AZIMUTH ON TARGET CURRENT AND HULL
HULL IS ASSUMED TO BE POINTING IN THE DIRECTION OF TRAVEL
CURRENT IS ASSUMED TO BE POINTING TOWARDS AN INTENDED
TARGET OR IF THIS TARGET VEHICLE IS NOT ENGAGING, THEN TURRET
IS STILL POINTING TOWARDS LAST TARGET.
IF TARGET VEHICLE IS IN SURVEILLANCE MODE, TURRET IS
IN LINE WITH THE HULL

IF (KSIDE.EQ.1) THEN
  AZMT=IRSA(J,2)
  AZS=AZTB(K)+180.0
  AZTT=AZTR(J)
ELSE
  AZMT=ISBA(J,2)
  AZS=AZTR(K)+180.0
  AZTT=AZTB(J)
ENDIF

AZH=AZS-AZMT
AZT=AZS-AZTT

IF (AZH.GE.360.0) AZH=AZH-360.0
IF (AZH.LT.0.0) AZH=AZH+360.0
IF (AZT.GE.360.0) AZT=AZT-360.0
IF (AZT.LT.0.0) AZT=AZT+360.0

IAZH=NINT(AZH)
IAZT=NINT(AZT)
IAZH=IAZH-360
IAZT=IAZT-360
IRNG=NINT(RANGE*1000.)

C FIND SHOOTER AZIMUTH TO TARGET RELATIVE TO SHOOTER HULL FRONT

IF (KSIDE.EQ.1) THEN
  AZHS=ISBA(K,2)
  AZTS=AZTB(K)
ELSE
  AZHS=IRSA(K,2)
  AZTS=AZTR(K)
ENDIF

IF (AZTS.LT.AZHS) AZTS=AZTS+360.0
AZOUT=AZTS-AZHS

IF (AZOUT.GE.360.0) AZOUT=AZOUT-360.0
IF (AZOUT.LT.0.0) AZOUT=AZOUT+360.0

IAZOUT=NINT(AZOUT)
IF (IAZOUT.GE.360) IAZOUT=IAZOUT-360

1 FORMAT (14H)
WRITE (11,1) KTIME,KSIDE,KJSYS,KWP,IENERGY,IRNG,JJSYS,
& IFAPS,KILL,IAZH,IAZT,IAZOUT

RETURN
END

SUBROUTINE OUTPUT2 (KTIME,JSIDE,J,KILL)
C WRITE OUTPUT DATA FOR POST PROCESSING ON SYSTEM STATUS WHEN
C KILLED OR AT END OF SIMULATION
C
C OUTPUT 2:
KTIME KILL TIME
JSIDE SYSTEM SIDE (1 BLUE, 2 RED)
J VEHICLE NUMBER
JSYS SYSTEM NUMBER
JSPD SPEED
NWP NUMBER OF WEAPONS
JAM1-3 REMAINING AMMUNITION EACH WEAPON
JENERGY BLUE EM GUN ENERGY (MJ) REMAINING
MODEF SYSTEM FIRE MODE
MODEM SYSTEM MOVEMENT MODE
JDEF DEFILADE (1) OR EXPOSED (0)
KILL TARGET KILLED FLAG (1 - ALIVE, 0 - KILLED)

COMMON/MOVE/BXYA(200,2),RXYA(200,2),ISBA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),

```

```

& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
& TRMFR(200),TRDES(200),TRDESR(200)
COMMON/BATTLE/ITIME,ITIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)
COMMON/DEFILADE/IDEFB(200),IDEFR(200)
COMMON/SHOTS/EMB(200,3),NSHOTB(200,3),NSHOTR(200,3)
COMMON/ENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
& ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2),
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRNDB(30),NRNDR(30),TEMAXB(30),TEMAXR(30),TEMINB(30),
& TEMINR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MILB(10,3),MILR(10,3)

CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*9 BFNAME,RFNAME

JAM1=0
JAM2=0
JAM3=0

IF (JSIDE.EQ.1) THEN
  JSYS=IRSYS(J,1)
  JSPD=IRSA(J,1)
  NWP=NWB(JSYS)
  JAM1=NSHOTB(J,1)
  IF (NWP.GE.2) JAM2=NSHOTB(J,2)
  IF (NWP.EQ.3) JAM3=NSHOTB(J,3)
  JENERGY=NINT(ELSYS(J))
  MODEF=MODEFB(J)
  MODEM=MODEMB(J)
  JDEF=IDEFB(J)
ELSE
  JSYS=IRSYS(J,1)
  JSPD=IRSA(J,1)
  NWP=NRW(JSYS)
  JAM1=NSHOTR(J,1)
  IF (NWP.GE.2) JAM2=NSHOTR(J,2)
  IF (NWP.EQ.3) JAM3=NSHOTR(J,3)
  JENERGY=0
  MODEF=MODEFR(J)
  MODEM=MODEMR(J)
  JDEF=IDEFR(J)
ENDIF

1 FORMAT (14H)

IF (JSYS.GT.10) GO TO 10
WRITE (12,1) KTIME,JSIDE,JJSYS,JSPD,NWP,JAM1,JAM2,JAM3,JENERGY,
& MODEF,MODEM,JDEF,KILL

10 CONTINUE
RETURN
END

SUBROUTINE WEND
C
C WRITES FINAL STATUS OF VEHICLES WHICH SURVIVED
C
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/BATTLE/ITIME,ITIME,ITGTB(200,2),ITGTR(200,2),
& PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
& IAMB(200)

CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

DO 100 J=1,NB
IF (IRSYS(J,2).EQ.1) CALL OUTPUT2 (ITIME,1,J,1)
100 CONTINUE

DO 200 J=1,NR
IF (IRSYS(J,2).EQ.1) CALL OUTPUT2 (ITIME,2,J,1)
200 CONTINUE

RETURN
END

SUBROUTINE TURAZ (KSIDE,K,J,AZIMUTH)
C SETS TURRET AZIMUTH TO CURRENT TARGET AZIMUTH

COMMON/MOVE/BXYA(200,2),RXYA(200,2),ISBA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM

CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

PI=3.14159

IF (KSIDE.EQ.1) THEN
  XS=BXYA(K,1)
  YS=BXYA(K,2)

```

```

      XT=RXYA(J,1)
      YT=RXYA(J,2)
    ELSE
      XS=RXYA(K,1)
      YS=RXYA(K,2)
      XT=BX YA(J,1)
      YT=BX YA(J,2)
    ENDF
  TOP=YT-YS
  BOT=XT-XS
  IF (BOT.EQ.0.0) BOT=BOT+.001
  DIR=ATAN(ABS(TOP)/ABS(BOT))*180./PI
  IF (BOT.LT.0.0) DIR=DIR+180.
  IF (TOP.GT.0.0) DIR=180.-DIR
  ELSE
    IF (TOP.LT.0.0) DIR=360.-DIR
  ENDF
  CALL FAZ (DIR,AZIMUTH)

RETURN
END

SUBROUTINE WMODES
  WRITES MODE TIMES FOR EACH SIXTY CYCLE PERIOD ONLY
  FOR BLUE SYSTEM #1

  COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
  & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,N8,NR
  COMMON/MODE/SMOKE/NB(200),MODEMR(200),TCMMB(200,3),
  & TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
  & MODEFR(200),CEWB(200),CEWR(200),TCFMB(200,8),TCFMR(200,8),
  & TTFMB(200,8),TTFMR(200,8),TRFMB(200),TRFMR(200),
  & TRMFR(200),TROESB(200),TROESR(200)
  COMMON/BATTLE/ITIME,IDTIME,ITGTR(200,2),ITGTR(200,2),
  & PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
  & IAMB(200)
  COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
  & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
  COMMON/ESMOKE/TESMK(200)

  CHARACTER*16 BHIER,RHIER,IHIER
  CHARACTER*24 BUNIT,RUNIT
  CHARACTER*30 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

  C
  1 FORMAT (5H)
  IDLT=80*IDTIME
  DO 100 K=1,NB
    IF (IBSYS(K,2).EQ.1) THEN
      KSYS=IBSYS(K,1)
      IF (KSYS.EQ.1) THEN
        DO 50 MM=1,3
          WRITE (13,1) ITIME,IDLT,K,MM,NINT(TCMMB(K,MM))
        DO 60 MEM=1,2
          WRITE (13,1) ITIME,IDLT,K,MEM,NINT(TEM(K,MEM))
        DO 70 MF=1,9
          WRITE (13,1) ITIME,IDLT,K,MF,NINT(TCFMB(K,MF))
        C ENGINE SMOKE DUTY CYCLE TIME
          WRITE (13,1) ITIME,IDLT,K,0,NINT(TESMK(K))
        ENDF
      ENDF
    100 CONTINUE
    RETURN
  END

SUBROUTINE WOTHERS (BACEL)
  WRITES OTHER OUTPUTS FOR BLUE EM VEHICLE #1 EACH TIME
  STEP ... VEHICLE SPEED, ACCELERATION, EM ENERGY
  FOR BLUE SYSTEM #1

  COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
  & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,N8,NR
  COMMON/BATTLE/ITIME,IDTIME,ITGTR(200,2),ITGTR(200,2),
  & PKTB(200,2),PKTR(200,2),KILLB(200),KILLR(200),IAMR(200),
  & IAMB(200)
  COMMON/EMENERGY/NBLUE,NEMW,NEMWN(3),EWP(3,2),EG(5),
  & ELSYS(200),MODEEM(200),TE(200),TEM(200,2),TTEM(200,2)
  COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MYPEB(200),MYPER(200)
  & ,BXYP(200,2),RXP(200,2),ISPD8(200),ISPD8(200)
  COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
  & L1BM,L2M,L3M,L1RM
  COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)

  C
  DIMENSION BACEL(200)

  CHARACTER*16 BHIER,RHIER,IHIER
  CHARACTER*24 BUNIT,RUNIT
  CHARACTER*30 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

  C
  1 FORMAT (5H)
  DO 100 K=1,NB
    IF (IBSYS(K,2).EQ.1) THEN
      KSYS=IBSYS(K,1)
      IF (KSYS.EQ.1) THEN
        IF (KSYS.EQ.1) THEN
          & WRITE (14,1) ITIME,IDTIME,X,ISPD8(K,NINT(BACEL(K)),NINT(ELSYS(K))
          & ,IBSA(K,2),NINT(GRADEB(K))
        ENDF
      100 CONTINUE
    RETURN
  END

SUBROUTINE READCOOK
  C READS VEHICLE BURNING PROBABILITY TABLE AND INITIALIZES ARRAY

  COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
  & KBURNB(200),KBURNR(200)

  CHARACTER*30 LINE

  NCOOK=0
  DO 10 I=1,400
    XYCOOK(I,1)=1.
  10 XYCOOK(I,2)=1.
  DO 11 I=1,200
    KBURNB(I)=0
  11 KBURNR(I)=0

  303 FORMAT (A30)
  503 FORMAT (1X,A30)
  OPEN (1,FILE='COOKERS.IN')
  READ (1,303) LINE
  C WRITE ('503) LINE
  READ (1,303) LINE
  C WRITE ('503) LINE
  READ (1,303) LINE
  C WRITE ('503) LINE
  READ (1,303) LINE
  C WRITE ('503) LINE
  313 FORMAT (10F8.2)
  314 FORMAT (1X,10F8.2)
  READ (1,313) (PBURNB(K),K=1,10)
  C WRITE ('314) (PBURNB(K),K=1,10)
  READ (1,303) LINE
  C WRITE ('503) LINE
  READ (1,313) (PBURNR(K),K=1,10)
  C WRITE ('314) (PBURNR(K),K=1,10)

  CLOSE (1)
  RETURN
  END

SUBROUTINE COOKS (KSIDE,NUM)
  COMMON/COOKERS/NCOOK,XYCOOK(400,2),PBURNB(10),PBURNR(10),
  & KBURNB(200),KBURNR(200)
  COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
  & L1BM,L2M,L3M,L1RM
  COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
  & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,N8,NR

  C
  CHARACTER*16 BHIER,RHIER,IHIER
  CHARACTER*24 BUNIT,RUNIT
  CHARACTER*30 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

  PB=RND0

  IF (KSIDE.EQ.1) THEN
    NSYS=IBSYS(NUM,1)
    PBURN=PBURNB(NSYS)
    X=BX YA(NUM,1)
    Y=BX YA(NUM,2)
    IF (PBURN.GE.PB) THEN
      NCOOK=NCOOK+1
      XYCOOK(NCOOK,1)=X
      XYCOOK(NCOOK,2)=Y
      KBURNB(NUM)=1
    ENDF
  ELSE
    NSYS=IRSYS(NUM,1)
    PBURN=PBURNR(NSYS)
    X=RX YA(NUM,1)
    Y=RX YA(NUM,2)
    IF (PBURN.GE.PB) THEN
      NCOOK=NCOOK+1
      XYCOOK(NCOOK,1)=X
      XYCOOK(NCOOK,2)=Y
      KBURNR(NUM)=1
    ENDF
  ENDF

  RETURN
  END

SUBROUTINE COOKMASK (XS,YS,RMAX1,RMAX2,RMAX3)
  C FIND ANGLE AND RANGE TO ALL COOKERS WITHIN MAX RANGE OF
  C THIS SYSTEM SO THAT MASKING CAN BE EVALUATED

```

```

C      COMMON/MASKCOOK/COOKMSK(400,3),NCOOKMSK
COMMON/COOKERS/NCOOK,XCOORD(400,2),PBURNB(10),PBURNR(10),
& KBURNB(200),KBURNR(200)

C      XS,YS -- SHOOTER COORDINATES
RMAX1,2,3 -- MAX RANGE OF SYSTEM WEAPONS
COOKMSK(1,1) -- X COORD OF EACH COOKER
COOKMSK(1,2) -- Y COORD OF EACH COOKER
COOKMSK(1,3) -- RADIUS FOR EACH COOKER PUFF
NCOOKMSK -- NUMBER OF COOKERS WITHIN MAX RANGE

C      PI=3.14159
NCOOKMSK=0
IF (NCOOK.GT.0) THEN
C      CYCLE THROUGH ALL COOKERS TO FILL MASKING ARRAY
RMAX=0
IF (RMAX1.GT.RMAX) RMAX=RMAX1
IF (RMAX2.GT.RMAX) RMAX=RMAX2
IF (RMAX3.GT.RMAX) RMAX=RMAX3
RMAX=RMAX/1000.0
DO 20 I=1,NCOOK
  XD=XCOORD(I,1)
  YD=YCOORD(I,2)
  RADIUS=0.030
  RANGE=((XD-XS)**2+(YD-YS)**2)**.5
  IF ((RANGE-RADIUS).GT.RMAX) GO TO 20
  NCOOKMSK=NCOOKMSK+1
  COOKMSK(NCOOKMSK,1)=XD
  COOKMSK(NCOOKMSK,2)=YD
  COOKMSK(NCOOKMSK,3)=RADIUS
20 CONTINUE
  ENDOF
  RETURN
END

SUBROUTINE COOKSEE (XS,YS,XT,YT,AZIMUTH,JSKIP)
EVALUATES DUST MASKING BETWEEN SHOOTER AND TARGET

C      COMMON/MASKCOOK/COOKMSK(400,3),NCOOKMSK
COMMON/COOKERS/NCOOK,XCOORD(400,2),PBURNB(10),PBURNR(10),
& KBURNB(200),KBURNR(200)

C      COOKMSK(1,1) -- X COORD OF EACH COOKER
COOKMSK(1,2) -- Y COORD OF EACH COOKER
COOKMSK(1,3) -- RADIUS FOR EACH COOKER PUFF
NCOOKMSK -- NUMBER OF COOKERS WITHIN MAX RANGE

C      XS,YS SHOOTER REAL WORLD COORDINATES
XT,YT TARGET REAL WORLD COORDINATES
AZIN AZIMUTH TO TARGET
JSKIP FLAG TO SKIP THIS TARGET IF MASKED (=0 TO SKIP)

C      COMMON/ELEVA/CONT(20),IELEV(10000)
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT

DOUBLE PRECISION ANGLE,ANGLEI

PI=3.14159

IF (NCOOKMSK.GT.0) THEN
C      FIND SHOOTER AND TARGET GRID COORDINATES
TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
CENTER TO CENTER ABOUT THE SCALE.
RANGE=((YT-YS)**2+(XT-XS)**2)**.5
IXS=NINT(XS/SCALE)
IYS=NINT(YS/SCALE)
IXT=NINT(XT/SCALE)
IYT=NINT(YT/SCALE)
IF (IXS.LT.1) IXS=1
IF (IXT.LT.1) IXT=1
IF (IYS.LT.1) IYS=1
IF (IYT.LT.1) IYT=1
IF (IXS.GT.NX) IXS=NX
IF (IXT.GT.NX) IXT=NX
IF (IYS.GT.NY) IYS=NY
IF (IYT.GT.NY) IYT=NY

C      X1=IXS*SCALE
X2=IXT*SCALE
Y1=IYS*SCALE
Y2=IYT*SCALE
RANGE=((Y2-Y1)**2+(X2-X1)**2)**.5

C      FIND SHOOTER AND TARGET ELEVATION
IS=NY*(IXS-1)+IYS
SELEV=CONT(IELEV(IS))
IT=NY*(IXT-1)+IYT
TELEV=CONT(IELEV(IT))

C      ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
FOR VEHICLE HEIGHT
SELEV=SELEV+.002
TELEV=TELEV+.002

C      FIND ELEVATION ANGLE TO THIS TARGET
TOP=TELEV-SELEV
BOT=RRANGE
ANGLE=ATAN(TOP/BOT)

C      DO 200 I=1,NCOOKMSK
IF (JSKIP.EQ.0) GO TO 300

C      CYCLE THROUGH ALL COOKERS
TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
BRACKETING EACH PUFF WITHIN RANGE. DROP IF OUT OF ANGLE RANGE.
THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE PUFF.
DROP PUFF IF NOT HIGH ENOUGH

C      EVALUATE RANGE TO COOKER PUFF
XP=COOKMSK(I,1)
YP=COOKMSK(I,2)
WRITE (10,*) 'XP YP ', XP,YP
RADIUS=COOKMSK(I,3)
WRITE (10,*) 'RADIUS ',RADIUS
RP=((YP-YS)**2+(XP-XS)**2)**.5
WRITE (10,*) 'PUFF RANGE ',RP, ' TARGET RANGE ',RRANGE
IF ((RP-RADIUS).GT.RRANGE) GO TO 200

C      EVALUATE ANGLES TO COOKER PUFF
TOP=YP-YS
BOT=XP-XS
IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP/ABS(BOT)))*180./PI
IF (BOT.LT.0.0) THEN
  IF (TOP.LT.0.0) DIR=DIR+180.
  IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
  IF (TOP.LT.0.0) DIR=360.-DIR
ENDIF

C      CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (DIR,AZIN)
IF (RADIUS.LT.0.030) RADIUS=0.030
IF (RP.LT.RADIUS) RP=RADIUS
DELAZ=ASIN(RADIUS/RP)*180./PI
AZLFT=AZIN-DELAZ
AZRGT=AZIN+DELAZ
IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.0
IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
WRITE (10,*) 'AZLFT',AZLFT, ' AZRIGHT',AZRGT, ' AZIMUTH ',AZIMUTH
IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200

C      EVALUATE HEIGHT OF COOKER PUFF

C      FIND ELEVATION OF THE GROUND UNDER THIS PUFF
IXP=NINT(XP/SCALE)
IYP=NINT(YP/SCALE)
IF (IXP.LT.1) IXP=1
IF (IYP.LT.1) IYP=1
IF (IXP.GT.NX) IXP=NX
IF (IYP.GT.NY) IYP=NY
C      PX2=IXP*SCALE
PY2=IYP*SCALE
PRANGE=((PY2-Y1)**2+(PX2-X1)**2)**.5
IP=NY*(IXP-1)+IYP
PELEV=CONT(IELEV(IP))

C      ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
MAX PUFF HEIGHT IS 30 METERS (100 FEET)
HEIGHT=RADIUS
IF (RADIUS.GT.0.030) HEIGHT=0.030
PELEV=PELEV+HEIGHT

C      FIND ELEVATION ANGLE TO THIS DUST PUFF
TOP=PELEV-SELEV
BOT=RP
ANGLEI=ATAN(TOP/BOT)
WRITE (10,*) 'TARGET ELEVATION ANGLE ',ANGLE
WRITE (10,*) 'PUFF ELEVATION ANGLE ',ANGLEI
IF (ANGLEI.GT.ANGLE) JSKIP=0
200 CONTINUE

C      ENDOF
300 RETURN
END

SUBROUTINE READMINES
C      READS MINEFIELD DATA

COMMON/MINES/NMINES,XYMIN(100,2),WLMINE(100,2),NAZMINE(100,3)
COMMON/MINEKILLS/MINEK(200,2)
COMMON/MINEBREECH/MINEBEECH(200,2),MINEON(100)

CHARACTER*80 LINE

C      NMINES -- NUMBER OF MINEFIELDS
XYMIN -- X AND Y COORDINATES OF CENTER
WLMINE -- WIDTH AND LENGTH OF MINEFIELD
NAZMINE -- NUMBER OF MINES IN FIELD, AZIMUTH OF LENGTH,
SURFACE (1) OR BURIED (0)

C      DO 10 I=1,100
MINEON(I)=1
WLMINE(I,1)=0.
WLMINE(I,2)=0.

```

```

      NAZMINE(1,1)=0
      NAZMINE(1,2)=0
      NAZMINE(1,3)=0
      XYMINE(1,1)=1.
10  XYMINE(1,2)=1.
303  FORMAT (A80)
503  FORMAT (1X,A80)
      OPEN (1,FILE='MINES.IN')
      READ (1,303) LINE
C    WRITE ('.503) LINE
      READ (1,303) LINE
C    WRITE ('.503) LINE
      READ (1,303) LINE
C    WRITE ('.503) LINE
311  FORMAT (I8)
312  FORMAT (1X,I8)
      READ (1,311) NMINES
      IF (NMINES.GT.100) NMINES=100
      IF (NMINES.LT.0) NMINES=0
C    WRITE ('.312) NMINES
313  FORMAT (8I8)
314  FORMAT (1X,8I8)
      READ (1,303) LINE
C    WRITE ('.503) LINE
      READ (1,303) LINE
C    WRITE ('.503) LINE
      READ (1,303) LINE
C    WRITE ('.503) LINE
      DO 320 I=1,NMINES
C    READ (1,313) N,IX,IY,L,RW,IAZ,NUM,LAID
C    WRITE ('.314) N,IX,IY,L,RW,IAZ,NUM,LAID
      XYMINE(1,1)=REAL(IX)/1000.
      XYMINE(1,2)=REAL(IY)/1000.
      WLMINE(1,1)=REAL(RW)/1000.
      WLMINE(1,2)=REAL(LR)/1000.
      NAZMINE(1,1)=NUM
      NAZMINE(1,2)=IAZ
      NAZMINE(1,3)=LAID
320  CONTINUE
      CLOSE (1)
      DO 400 I=1,200
      MINEK(1,1)=0
      MINEK(1,2)=0
      MBREECH(1,1)=0
      MBREECH(1,2)=0
400  CONTINUE
      RETURN
      END
SUBROUTINE PLOTMINES
C C C C C
PLOTS MINEFIELD LOCATIONS ON MAP
FILLS THE AREA IF THE MINEFIELD IS SURFACE LAID OR BURIED
BUT "DISCOVERED" BY A VEHICLE GETTING BLASTED
IF UNDISCOVERED AND BURIED, THE AREA IS NOT FILLED IN
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)
C C C C C
DIMENSION XARRAY(4),YARRAY(4)
C C C C C
PI=3.14159
C C C C C
DRAWS AND FILLS A SQUARE FOR SURFACE OR KNOWN MINEFIELD
CALL FACTOR (ZFACT)
GXUNIT=11.0/NX*SCFACT
GYUNIT=8.5/NY
XUNIT=GXUNIT/SCALE
YUNIT=GYUNIT/SCALE
C    DO 200 I=1,NMINES
C    IF (NAZMINE(I,3).EQ.0) GO TO 200
C    IF (MINEON(I).EQ.0) GO TO 200
      XC=XYMINE(I,1)
      YC=XYMINE(I,2)
      WIDTH=WLMINE(I,1)
      WLONG=WLMINE(I,2)
      AZ=REAL(NAZMINE(I,2))
      CALL FDIR (AZ,DIR)
      DIR=DIR/180.*PI
      X1=XC-WLONG/2.*COS(DIR)-WIDTH/2.*SIN(DIR)
      Y1=YC-WLONG/2.*SIN(DIR)+WIDTH/2.*COS(DIR)
      X2=X1+WIDTH*SIN(DIR)
      Y2=Y1-WIDTH*COS(DIR)
      X3=X2-WLONG*COS(DIR)
      Y3=Y2-WLONG*SIN(DIR)
      X4=X3-WIDTH*SIN(DIR)
      Y4=Y3+WIDTH*COS(DIR)

```

```

      IF (XLEN.GT.0.) CALL FACTOR (ZFACTN)
      ISKIP=0
      CALL NEWPEN (13)
      XARRAY(1)=X1*XUNIT
      XARRAY(2)=X2*XUNIT
      XARRAY(3)=X3*XUNIT
      XARRAY(4)=X4*XUNIT
      YARRAY(1)=Y1*YUNIT
      YARRAY(2)=Y2*YUNIT
      YARRAY(3)=Y3*YUNIT
      YARRAY(4)=Y4*YUNIT
      Y10=.95*YMAX
      X10=.95*XMAX
      X11=.05*XMAX
      Y11=.05*YMAX
      DO 100 J=1,4
      IF (XARRAY(J).LE.X11) XARRAY(J)=X11
      IF (YARRAY(J).LE.Y11) YARRAY(J)=Y11
      IF (XLEN.GT.0.) THEN
        CALL ZOOMIT (XARRAY(J),YARRAY(J),ISKIP)
        IF (ISKIP.EQ.1) THEN
          IF (YARRAY(J).GT.Y10) YARRAY(J)=Y10
          IF (XARRAY(J).GT.X10) XARRAY(J)=X10
          ISKIP=0
        ENDIF
      ENDIF
100  CONTINUE
      IF (XLEN.GT.0.) THEN
      IF (XARRAY(1).LE.X11.AND.XARRAY(2).LE.X11.AND.XARRAY(3).LE.X11.
      & .AND.XARRAY(4).LE.X11) ISKIP=1
      IF (YARRAY(1).LE.Y11.AND.YARRAY(2).LE.Y11.AND.YARRAY(3).LE.Y11.
      & .AND.YARRAY(4).LE.Y11) ISKIP=1
      IF (XARRAY(1).GE.X10.AND.XARRAY(2).GE.X10.AND.XARRAY(3).GE.X10.
      & .AND.XARRAY(4).GE.X10) ISKIP=1
      IF (YARRAY(1).GE.Y10.AND.YARRAY(2).GE.Y10.AND.YARRAY(3).GE.Y10.
      & .AND.YARRAY(4).GE.Y10) ISKIP=1
      ENDIF
      IF (ISKIP.EQ.1) GO TO 200
      IF (NAZMINE(I,3).EQ.1) THEN
C    DRAW AND FILL FOUR SIDED SQUARE
      CALL FILL (4,XARRAY,YARRAY)
      ENDIF
C    IF (NAZMINE(I,3).EQ.0) THEN
C    DRAW EMPTY BOX
      CALL PLOT (XARRAY(1),YARRAY(1),3)
      CALL PLOT (XARRAY(2),YARRAY(2),2)
      CALL PLOT (XARRAY(3),YARRAY(3),2)
      CALL PLOT (XARRAY(4),YARRAY(4),2)
      CALL PLOT (XARRAY(1),YARRAY(1),2)
      ENDIF
200  CONTINUE
C    READ ('.')
      RETURN
      END
SUBROUTINE MINEFIELD (KSIDE,K,DELPOS,DIR,INNOW,INNEXT)
COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
COMMON/MINEBREECH/MBREECH(200,2),MINEON(100)
COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),RSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,L2RF,L3RF,NB,NR
C    CHARACTER*16 BHIER,RHIER,IBIER
C    CHARACTER*24 BUNIT,RUNIT
C    CHARACTER*90 L1BF,L2F,L3F,L1RF,L2RF,L3RF,L1BM,L2M,L3M,L1RM
C C C C C
C    KSIDE UNIT SIDE
C    K UNIT NUMBER
C    DELPOS DISTANCE TO BE TRAVELED IN THIS TIME STEP
C    DIR TRAVEL DIRECTION (CARTESIAN ANGLE)
      PI=3.14159
      INNOW=0
      INNEXT=0
      IF (KSIDE.EQ.1) THEN
        XK=BXYA(K,1)
        YK=BXYA(K,2)
      ELSE
        XK=RXYA(K,1)
        YK=RXYA(K,2)
      ENDIF
      DO 100 I=1,NMINES
      IF (MINEON(I).EQ.0) GO TO 100
      IF (INNOW.NE.0.OR.INNEXT.NE.0) GO TO 200
      XC=XYMINE(I,1)
      YC=XYMINE(I,2)
      WIDTH=WLMINE(I,1)

```

```

WLONG=WLMI(1,2)
AZ=REAL(NAZMI(1,2))
CALL FDIR (AZ,08L)

C SEE IF THIS UNIT IS CURRENTLY IN THIS MINEFIELD

C FIND RANGE TO CENTER OF THE MINEFIELD
RC=(YC-YK)**2+(XC-XK)**2**5
C FIND RANGE ANGLE TO CENTER OF MINEFIELD
TOP=YC-YK
BOT=XC-XK
IF (BOT.EQ.0.0) BOT=BOT+.001
RDIR=ATAN(ABS(TOPYABS(BOT)))*180./PI
IF (BOT.LT.0.0) THEN
    IF (TOP.LT.0.0) RDIR=RDIR+180.
    IF (TOP.GT.0.0) RDIR=180.-RDIR
ELSE
    IF (TOP.LT.0.0) RDIR=360.-RDIR
ENDIF

C FIND LINE-OF-SIGHT THICKNESS OF THIS MINEFIELD TOWARDS UNIT
ALPHA=ABS(OBL-90.-RDIR)
IF (ALPHA.EQ.90.0.OR.ALPHA.EQ.270.) THEN
    TLOS=WLONG/2.
ELSE
    TLOS=(ABS(WIDTH/COS(ALPHA/180.*PI)))/2.
    IF (TLOS.GT.(WLONG/2.)) TLOS=WLONG/2.
ENDIF

IF (TLOS.GT.RC) INNOW=1
C THIS UNIT IS CURRENTLY IN THIS MINEFIELD

C FIND IF THIS UNIT WILL BE IN THE FIELD IF IT CONTINUES
C TO MOVE IN CURRENT DIRECTION
XK=XK+DELPOS*COS(DIR*PI/180.)
YK=YK+DELPOS*SIN(DIR*PI/180.)

C FIND RANGE TO CENTER OF THE MINEFIELD
RC=(YC-YK)**2+(XC-XK)**2**5
C FIND RANGE ANGLE TO CENTER OF MINEFIELD
TOP=YC-YK
BOT=XC-XK
IF (BOT.EQ.0.0) BOT=BOT+.001
RDIR=ATAN(ABS(TOPYABS(BOT)))*180./PI
IF (BOT.LT.0.0) THEN
    IF (TOP.LT.0.0) RDIR=RDIR+180.
    IF (TOP.GT.0.0) RDIR=180.-RDIR
ELSE
    IF (TOP.LT.0.0) RDIR=360.-RDIR
ENDIF

C FIND LINE-OF-SIGHT THICKNESS OF THIS MINEFIELD TOWARDS UNIT
ALPHA=ABS(OBL-90.-RDIR)
IF (ALPHA.EQ.90.0.OR.ALPHA.EQ.270.) THEN
    TLOS=WLONG/2.
ELSE
    TLOS=(ABS(WIDTH/COS(ALPHA/180.*PI)))/2.
    IF (TLOS.GT.(WLONG/2.)) TLOS=WLONG/2.
ENDIF

IF (TLOS.GT.RC) INNEXT=1
C THIS UNIT WILL REMAIN IN OR ENTER THIS MINEFIELD

100 CONTINUE
200 CONTINUE

RETURN
END

SUBROUTINE MINEKILL (KSIDE,K,INNOW,INNEXT,DELPOS)
COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
COMMON/MINEKILLS/MINEK(200,2)

IF (INNOW.NE.0) THEN
    AFIELD=WLMINE(INNOW,1)*WLMINE(INNOW,2)
    DENS=REAL(NAZMINE(INNOW,1))/AFIELD
    IFOUND=INNOW
ELSE
    IF (INNEXT.NE.0) THEN
        AFIELD=WLMINE(INNEXT,1)*WLMINE(INNEXT,2)
        DENS=REAL(NAZMINE(INNEXT,1))/AFIELD
        IFOUND=INNEXT
    ENDIF
ENDIF

IF (IFOUND.NE.0) THEN
    AREA=DELPOS*.005
    PMU=DENS*AREA
    P1=EXP(-PMU)
    PLVE=RND0
    IF (P1.GT.PLVE) MINEK(K,KSIDE)=IFOUND
    BOOM .....
ENDIF

RETURN
END

```

# SUBROUTINE WMINE (KTIME,JSIDE,J,NFIELD)

```

CCCCCCCCC MINEFIELD KILLS DATA OUTPUT
KTIME      KILL TIME
JSIDE      SYSTEM SIDE (1 BLUE, 2 RED)
J          VEHICLE NUMBER
JSYS       SYSTEM NUMBER
NFIELD     MINEFIELD NUMBER

COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
CHARACTER*16 BHIER,RHIER,IHIER
CHARACTER*24 BUNIT,RUNIT

IF (JSIDE.EQ.1) THEN
    JSYS=IBSYS(J,1)
ELSE
    JSYS=IRSYS(J,1)
ENDIF

1 FORMAT (5(4,3X))

WRITE (15,1) KTIME,JSIDE,J,JSYS,NFIELD

RETURN
END

SUBROUTINE BREECH
CCCCCCCCC PLACES VEHICLES IN MINEFIELD BREECH MODE WHICH FIXES
THEIR SPEED TO 4 KM/HR, BUT MAKES THEM INVULNERABLE TO MINES.
THIS ROUTINE ALSO ALLOWS THE OPERATOR TO TURN MINEFIELDS
ON OR OFF TO SIMULATE BREECHED MINEFIELDS OR UNITS MOVING
QUICKLY THROUGH SAFE LANES IN KNOWN MINEFIELDS.

COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXYA(200,2),BSA(200,2),RSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/WEAPONS/NBF,NRF,BFNAME(10),RFNAME(10),NWB(10),NWR(10),
& NRWB(10,3),NRWR(10,3),RB(180),RR(180),PKB(180,20),PKR(180,20),
& NRND(30),NRNR(30),TEMAB(30),TEMAR(30),TEMIB(30),
& TEMIR(30),ACOB(10,5),ACOR(10,5),SRCHB(10),SRCHR(10),
& MILB(10,3),MILR(10,3)
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& BXYP(200,2),RXYP(200,2),ISPB(200),ISPR(200)
COMMON/MODES/MODEMB(200),MODEMR(200),TCMMB(200,3),
& TCMMR(200,3),TTMMB(200,3),TTMMR(200,3),MODEFB(200),
& MODEFR(200),KEWB(200),KEWR(200),TCFMB(200,9),TCFMR(200,9),
& TTFMB(200,9),TTFMR(200,9),TRMMB(200),TRMFB(200),TRMMR(200),
& TRMFR(200),TRDESB(200),TRDESR(200)
COMMON/MINES/NMINES,XYMINE(100,2),WLMINE(100,2),NAZMINE(100,3)
COMMON/MINEKILLS/MINEK(200,2)
COMMON/MINEBREECH/MBBREECH(200,2),MINEON(100)

C CHARACTER*8 BFNAME,RFNAME
C CHARACTER*80 LINE
C CHARACTER*16 BHIER,RHIER,IHIER
C CHARACTER*24 BUNIT,RUNIT
C CHARACTER*80 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C CHARACTER*1 C2
C CHARACTER*1 C2,C3,C4,C5,C6,C7,C8,C9,C10,BLANK

2 WRITE (":")
WRITE (":") 1 -- TURN MINEFIELDS ON AND OFF
WRITE (":") 2 -- PLACE VEHICLES IN BREECH MODE
WRITE (":") 3 -- QUIT
WRITE (":")
READ (":",ERR=2) IC1
IF (IC1.EQ.3) GO TO 1000
IF (IC1.EQ.2) GO TO 10
3 IF (IC1.EQ.1) THEN
    WRITE (":") FIELD # ON/OFF Xcenter Ycenter
    4 FORMAT (1X,2(15,4X),2(F8.3,1X))
    DO 5 I=1,NMINES
        5 WRITE (":") I,MINEON(I),XYMINE(I,1),XYMINE(I,2)
        WRITE (":")
        WRITE (":") 'ENTER FIELD # (0=QUIT), OFF=0, ON=1'
        WRITE (":")
        READ (":",ERR=3) I,MINEON(I)
        IF (I.EQ.0) GO TO 2
        GO TO 3
    ENDIF
    GO TO 2

CCCCCCCCC BREECH MODE
10 KSIDE=0
WRITE (":")
WRITE (":") 'enter BLUE or RED, or QUIT'
WRITE (":")
READ (":",ERR=10) C2

```



```

      IF (C2(1:1).EQ.8) KSIDE=1
      IF (C2(1:1).EQ.9) KSIDE=2
      IF (C2(1:1).EQ.0) GO TO 2
      IF (KSIDE.EQ.0) GO TO 10
C
6  WRITE (C,*)
   WRITE (C,*) 'CURRENT BREECH STATUS'
   WRITE (C,*) 'UNIT HIERARCHY'  IDENTITY  1 = YES
   & 0 = NO -1 = RECKLESS
   IF (C2(1:1).EQ.8) THEN
C  BLUE
      KK=0
      DO 130 K=1,NB
121  FORMAT (1X,I3,1X,A16,A24,7X,I3)
      IF (IBSYS(K,2).GE.1) THEN
        WRITE (C,121) K,BHIER(K),BUNIT(K),MBREECH(K,1)
        KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
        READ (C,*)
        KK=0
      ENDIF
130  CONTINUE
C  ELSE
C  RED
      KK=0
      DO 140 K=1,NR
      IF (IRSYS(K,2).GE.1) THEN
        WRITE (C,121) K,RHIER(K),RUNIT(K),MBREECH(K,2)
        KK=KK+1
      ENDIF
      IF (KK.EQ.20) THEN
        READ (C,*)
        KK=0
      ENDIF
140  CONTINUE
      ENDIF
141  WRITE (C,*) 'ENTER COMMAND UNIT (0-RELISTS -1-QUIT)'
      WRITE (C,*)
      READ (C,*) ERR=141) ICOMMAND
      IF (ICOMMAND.EQ.0) GO TO 8
      IF (ICOMMAND.EQ.-1) GO TO 10
142  WRITE (C,*) 'ENTER NUMBER RANGE OF SUBORDINATES TO FOLLOW'
      WRITE (C,*) 'THIS ORDER (#START #END) (0 0 FOR NONE)'
      WRITE (C,*)
      READ (C,*) ERR=142) NSUB1,NSUB2
      IF (NSUB1.LE.0) THEN
        NSUB1=0
        NSUB2=0
      ENDIF
      IF (NSUB2.LE.0) THEN
        NSUB1=0
        NSUB2=0
      ENDIF
C
150  WRITE (C,*)
      WRITE (C,*) 'TO BREECH MODE THESE VEHICLES ENTER 1, ELSE 0'
      WRITE (C,*) 'RECKLESS BREECH (NO STOPPING OR CLEARING) ENTER -1'
      WRITE (C,*) 'FOR 0 AND -1, ENTER DESIRED SPEED AT MOVEMENT ORDER'
      &COMMAND LEVEL
      WRITE (C,*)
      READ (C,*) ERR=150) JDEF
      IF (JDEF.LT.-1.OR.JDEF.GT.1) GO TO 150

      IF (KSIDE.EQ.1) THEN
        IBSA(ICOMMAND,1)=4
        ISPDB(ICOMMAND)=4
        MBREECH(ICOMMAND,1)=JDEF
        MODEMB(ICOMMAND)=1
      ELSE
        IRSA(ICOMMAND,1)=4
        ISPDR(ICOMMAND)=4
        MBREECH(ICOMMAND,2)=JDEF
        MODEMR(ICOMMAND)=1
      ENDIF
      DO 200 I=NSUB1,NSUB2
      IF (KSIDE.EQ.1) THEN
        IF (IBSYS(I,2).GE.1) THEN
          IBSA(I,1)=4
          ISPDB(I)=4
          MBREECH(I,1)=JDEF
          MODEMB(I)=1
        ENDIF
      ELSE
        IF (IRSYS(I,2).GE.1) THEN
          IRSA(I,1)=4
          ISPDR(I)=4
          MBREECH(I,2)=JDEF
          MODEMR(I)=1
        ENDIF
      ENDIF
200  CONTINUE
      GO TO 141
1000 CONTINUE
      RETURN
      END

```

```

SUBROUTINE PKMOVE (KSIDE,IS,IT,RNG,BORE,TOF,PK)
C
C MOVING
C REDUCES PK WHEN ENGAGING MOVING TARGETS AND WHEN SHOOTER IS
C FIRE-AND-FORGET WEAPONS ONLY.
C TWO ERROR SOURCES DOMINATE THIS PROBLEM:
C 1) THE ACCURACY OF TURRET TRAVERSE RATE SENSORS (ASSUMED
C 1% ERROR);
C 2) THE MOVEMENT OF THE TARGET UP OR DOWN A GRADE.
C THE FIRST CONDITION RESULTS IN IMPROPER LEAD ANGLE.
C THE SECOND RESULTS IN WRONG SUPER-ELEVATION OF THE GUN.
C ALL OTHER MOVEMENTS OF THE SHOOTER AND TARGET
C ARE COMPENSATED WELL ENOUGH TO ASSUME NO ERROR.
C
COMMON/MOVE/BSYA(200,2),RXYA(200,2),ISBA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/ACEL/ACELB(10,5),ACELR(10,5),GRADEB(200),GRADER(200)
COMMON/MAEUVER/BSYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
& ,BSYP(200,2),RXP(200,2),ISPOB(200),ISPOR(200)

DIMENSION SIG(40),AREA(40)
CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
DATA SIG /0,1,2,3,4,5,6,7,8,9,1,1,1,2,3,4,1,5,
& 1,6,1,7,1,8,1,9,2,2,1,2,2,2,3,2,4,2,5,2,6,2,7,2,8,2,9,3,3,1,
& 3,2,3,3,4,3,5,3,6,3,7,3,8,3,9/
DATA AREA /5,5398,5793,6179,6564,6915,7258,758,7881,
& .8159,8413,8643,8849,9032,9192,9322,9452,9564,9641,
& .9713,9772,9821,9861,9893,9918,9938,9953,9965,9974,
& .9981,9987,9990,9993,9995,9997,9998,9998,9999,9999,
& 1.000/

PI=3.14159

RANGE=RNG*1000.
C
C IF (KSIDE.EQ.1) THEN
C  BLUE SHOOTER / RED TARGET
C  SS=REAL(ISPOB(1))/3600.*1000.
C  ST=REAL(ISPOR(1))/3600.*1000.
C  CALL FDIR (REAL(ISBA(1,2)),AZS)
C  CALL FDIR (REAL(IRSA(1,2)),AZT)
C  AZS=AZS/180.*PI
C  AZT=AZT/180.*PI
C  XS=BSYA(1,1)
C  YS=BSYA(1,2)
C  XT=RXYA(1,1)
C  YT=RXYA(1,2)
C  GRADET=GRADER(1)
C  ELSE
C  SS=REAL(ISPOB(1))/3600.*1000.
C  ST=REAL(ISPOR(1))/3600.*1000.
C  CALL FDIR (REAL(IRSA(1,2)),AZS)
C  CALL FDIR (REAL(ISBA(1,2)),AZT)
C  AZS=AZS/180.*PI
C  AZT=AZT/180.*PI
C  XS=RXYA(1,1)
C  YS=RXYA(1,2)
C  XT=BSYA(1,1)
C  YT=BSYA(1,2)
C  GRADET=GRADER(1)
C  ENDIF
C
C  DEFINE UNIT VECTOR CONNECTING SHOOTER AND TARGET
C  RMAG=((XS-XT)**2+(YS-YT)**2)**.5
C  DX=XS-XT
C  DY=YS-YT
C  U1=DX/RMAG
C  U2=DY/RMAG
C
C  FIND RELATIVE CROSSING SPEED OF SHOOTER AND TARGET
C  BY TAKING CROSS PRODUCT OF EACH VEHICLE VELOCITY VECTOR
C  ONTO THIS UNIT VECTOR. THEN TAKE DIFFERENCE IN CROSS
C  PRODUCTS.
C
C  AS=SS*COS(AZS)
C  BS=SS*SIN(AZS)
C  AT=ST*COS(AZT)
C  BT=ST*SIN(AZT)
C
C  THE CROSS PRODUCT OF THESE VECTOR QUANTITIES IS THE CROSSING
C  SPEED
C
C  SCROSS=ABS((AS*U1-BT*U2)-(AT*U2-BT*U1))
C
C  THIS CROSSING SPEED WILL HAVE A 1% ERROR TO GIVE DEFLECTION
C  AIMPOINT ERROR WHEN BULLET ARRIVES AT TARGET RANGE.
C
C  FIND TIME TO TARGET RANGE
C  TT=RANGE/BORE*TOF
C
C  FIND HORIZONTAL DISTANCE ERROR TO TARGET (METERS)
C  DHT=TT*SCROSS*.01
C
C  THIS BECOMES THE AIMPOINT OFFSET DISTANCE (METERS)
C  DIST=DHT
C
C  ASSUME WEAPON PK/H=1.0 AND THEREFORE HIT PROBABILITY IS
C  EQUAL TO PK/S AS ENTERED ABOVE
C  FIND DEFLECTION AND RANGE ERROR STANDARD DEVIATIONS
C  DAREA=PK*.5
C  RAREA=DAREA

```





```

10      ENDF
      CONTINUE
      RETURN
      ENDF

C      SUBROUTINE IDFIREORS
C      ENTER INDIRECT-FIRE ORDERS FOR ARTILLERY UNITS
C      CHARACTER*1 C2
10      KSIDE=0
      WRITE (C2) 'ARTILLERY FIRE MISSION ORDERS'
      WRITE (C2) 'enter BLUE or RED, or QUIT'
      READ (C2,ERR=10) C2
      IF (C2(1:1).EQ.'B') KSIDE=1
      IF (C2(1:1).EQ.'R') KSIDE=2
      IF (C2(1:1).EQ.'Q') GO TO 1000
      IF (KSIDE.EQ.0) GO TO 10
C      WRITE (C2)
C      WRITE (C2) '1 - LIST CURRENT MISSION BACKLOG'
C      WRITE (C2) '2 - LIST AVAILABLE ARTILLERY UNITS'
C      WRITE (C2) '3 - LIST AVAILABLE TARGETS'
C      WRITE (C2) '4 - MATCH A TARGET WITH ARTILLERY UNIT RANGES'
C      WRITE (C2) '5 - REQUEST A FIRE MISSION'
C      WRITE (C2) '6 - QUIT'
      READ (C2,ERR=5) MISSION
      IF (MISSION.LT.0.OR.MISSION.GT.6) GO TO 5
      IF (MISSION.EQ.6) GO TO 10
      IF (MISSION.EQ.1) CALL BACKLOG (KSIDE)
      IF (MISSION.EQ.2) CALL ARTAVAIL (KSIDE)
      IF (MISSION.EQ.3) CALL ARTTGT (KSIDE)
      IF (MISSION.EQ.4) CALL ARTTRNG (KSIDE)
      IF (MISSION.EQ.5) CALL ARTMISH (KSIDE)
      GO TO 5
1000   CONTINUE
      RETURN
      ENDF

C      SUBROUTINE BACKLOG (KSIDE)
C      DISPLAYS CURRENT ARTILLERY FIRE MISSION BACKLOG
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
      & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
      & XYFUR(200,2),MLAB(200,10),MLAR(200,10)
      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
      & NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5),
      & IAFAB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
      & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
      & ISCOOTB(10,4),ISCOOTR(10,4)
      CHARACTER*9 BANAME,RANAME
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      WRITE (C2)
      WRITE (C2) 'FIRE MISSION BACKLOG'
      IF (KSIDE.EQ.1) THEN
        WRITE (C2) 'BLUE'
        NM=NFMIS(1)
      ELSE
        WRITE (C2) 'RED'
        NM=NFMIS(2)
      ENDF
      WRITE (C2) 'MISS # UNIT # UNIT I.D. #RND RNDTYPE'
      & AIM XY AREA E.T.A'
      KK=0
      DO 20 K=1,NM
10      FORMAT (1X,I5,2X,I5,3X,A24,I6,1X,I6,1X,F7,3,1X,F7,3,1X,I6,1X,I6)
11      FORMAT (1X,10I5)
      IF (KSIDE.EQ.1) THEN
        IF (MFUB(K,1).EQ.0) GO TO 20
        WRITE (C2) K,MFUB(K,1),BUNIT(MFUB(K,1)),MFUB(K,2),MFUB(K,3),
      & XYFUB(K,1),XYFUB(K,2),MFUB(K,4),MFUB(K,5)
        KASYS=IRSYS(MFUB(K,1),1)-10
        IF (IAFAFB(KASYS,MFUB(K,3)).EQ.1) THEN
          WRITE (C2) 'GUIDED PROJECTILE TARGET LIST'
          WRITE (C2) (MLAB(K,J),J=1,10)
        ENDF
      ELSE
        IF (MFUR(K,1).EQ.0) GO TO 20
        WRITE (C2) K,MFUR(K,1),RUNIT(MFUR(K,1)),MFUR(K,2),MFUR(K,3),
      & XYFUR(K,1),XYFUR(K,2),MFUR(K,4),MFUR(K,5)
        KASYS=IRSYS(MFUR(K,1),1)-10
        IF (IAFAFR(KASYS,MFUR(K,3)).EQ.1) THEN

```

```

      WRITE (C2) 'GUIDED PROJECTILE TARGET LIST'
      WRITE (C2) (MLAR(K,J),J=1,10)
    ENDF
    ENDF
    KK=KK+1
    IF (KK.EQ.20) THEN
      READ (C2)
      KK=0
    ENDF
20   CONTINUE
      RETURN
      ENDF

      SUBROUTINE ARTAVAIL (KSIDE)
C      DISPLAYS CURRENTLY AVAILABLE ARTILLERY UNITS AND STATUS
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
      & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/AAVAIL/BAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT
      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*8 STATUS
C      AVAILABILITY FLAGS
C      1 - AVAILABLE FOR A FIRE MISSION
C      0 - UNAVAILABLE (CURRENTLY COMMITTED - SHOOTING)
C      -1 - UNAVAILABLE (SCOOTING)
C      9 - NOT AN ARTILLERY UNIT
      STATUS=' '
      WRITE (C2)
      WRITE (C2) 'ARTILLERY UNIT AVAILABILITY'
      IF (KSIDE.EQ.1) THEN
        WRITE (C2) 'BLUE'
      ELSE
        WRITE (C2) 'RED'
      ENDF
      WRITE (C2) 'UNIT # UNIT I.D. #TUBES AMMO BY #
      & STATUS'
      KK=0
      DO 20 K=1,200
10      FORMAT (1X,I5,2X,A24,I3,1X,I5,I3,1X,I6)
      IF (KSIDE.EQ.1) THEN
        IF (IRSYS(K,1).GT.10) THEN
          IF (IRAVL(K).EQ.1) STATUS='SCOOTING'
          IF (IRAVL(K).EQ.0) STATUS='SHOOTING'
          IF (IRAVL(K).EQ.-1) STATUS='SCOOTING'
          IF (IRSYS(K,2).LE.0) STATUS='KILLED'
          WRITE (C2) K,BUNIT(K),IRSYS(K,2),NRARTB(K,J),J=1,5),STATUS
          KK=KK+1
        ENDF
      ELSE
        IF (IRSYS(K,1).GT.10) THEN
          IF (IRAVL(K).EQ.1) STATUS='SCOOTING'
          IF (IRAVL(K).EQ.0) STATUS='SHOOTING'
          IF (IRAVL(K).EQ.-1) STATUS='SCOOTING'
          IF (IRSYS(K,2).LE.0) STATUS='KILLED'
          WRITE (C2) K,RUNIT(K),IRSYS(K,2),NRARTR(K,J),J=1,5),STATUS
          KK=KK+1
        ENDF
      ENDF
      IF (KK.EQ.20) THEN
        READ (C2)
        KK=0
      ENDF
20   CONTINUE
      RETURN
      ENDF

      SUBROUTINE ARTTGT (KSIDE)
C      FINDS AND DISPLAYS ALL POSSIBLE ARTILLERY TARGETS BASED ON
C      FORWARD OBSERVER AND AIRBORNE SURVEILLANCE ASSETS
C      AIRBORNE ASSETS GIVE POSITION, DIRECTION, AND SPEED OF ALL
C      THREAT VEHICLES, BUT THEY ARE UNIDENTIFIED.
C      THE FORWARD OBSERVER CAN IDENTIFY A VEHICLE BY SYSTEM TYPE
      COMMON/ARTTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
      & IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),ISSA(200,2),IRSA(200,2),
      & L1BM,L2M,L3M,L1RM
      COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPER(200)
      & BXPY(200,2),RXPY(200,2),ISPD(200),ISPD(200)
      COMMON/TRACKING/BOREB(10,3),BORE(10,3),IRNGB(30,4),IRNGR(30,4),
      & ITRKB(10,4),ITRKR(10,4),TOFB(10,3),TOFR(10,3)
      COMMON/FLAGS/ITER
      COMMON/AAVAIL/BAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)

```

```

INTEGER ARTGTB,ARTGTR
CHARACTER*10 BHER,PHER,INHER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM

C BLUE
IF (KSIDE.EQ.1) THEN
  WRITE (1,*)
  WRITE (1,*) RED UNITS TARGETABLE FOR BLUE ARTILLERY FIRE
  WRITE (1,*)
  IF (ITLB.EQ.0) THEN
    DO 10 I=1,NR
      ARTGTB(I,1)=0
      ARTGTB(I,2)=0
      IF (IRSYS(I,2).GE.1) ARTGTB(I,1)=1
10  CONTINUE
    DO 150 K=1,NB
      LOOP 1
      CHECK TO SEE IF THIS BLUE UNIT IS KILLED AND SKIP
      IF (IBSYS(K,2).EQ.0.OR.IBSYS(K,1).GT.10) GO TO 150
      FIND ALL RED TARGETS
      LOOP 3
      FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
      SO THAT MASKING CAN BE EVALUATED
      KSIDE=1
      MYSIDE=1
      IBBLUE=K
      CALL VMASK (KSIDE,IBBLUE)
      FIND ALL RED TARGETS OUT TO 6000 METERS RANGE
      NSYS=IBSYS(K,1)
      RMAX1=6000.
      GET X,Y COORDINATES FOR THIS BLUE UNIT
      XB=BXA(K,1)
      YB=BYA(K,2)
      IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
      FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
      EVALUATED
      ITRK1=ITRKB(NSYS,1)
      IF (ITRK1.EQ.0)
        & CALL DMASK (XB,YB,RMAX1,RMAX1,RMAX1)
      FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
      MASK VISUAL AND IR SYSTEMS
      CALL COOKMASK (XB,YB,RMAX1,RMAX1,RMAX1)
      IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
      FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
      CAN BE EVALUATED
      ITRK2=ITRKB(NSYS,2)
      IF (ITRK2.EQ.0)
        & CALL ESMASK (XB,YB,RMAX1,RMAX1,RMAX1)
      IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
      FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
      CAN BE EVALUATED
      ITRK3=ITRKB(NSYS,3)
      IF (ITRK3.EQ.0)
        & CALL GRMASK (XB,YB,RMAX1,RMAX1,RMAX1)
      IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
      ACQUISITION,
      FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
      MASKING CAN BE EVALUATED
      ITRK4=ITRKB(NSYS,4)
      IF (ITRK4.EQ.0)
        & CALL ASMASK (XB,YB,RMAX1,RMAX1,RMAX1)
      DO 140 J=1,NR
      LOOP 4
      CHECK TO SEE IF THIS RED UNIT IS KILLED OR ALREADY IDENTIFIED
      IF (IRSYS(J,2).EQ.0.OR.ARTGTB(J,2).EQ.1) GO TO 140
      FIND THIS RED UNIT'S X,Y COORDINATES
      XR=RXA(J,1)
      YR=RYA(J,2)
      FIND RANGE TO THIS TARGET
      RANGE=((XR-XB)**2+(YR-YB)**2)**.5
      CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
      NOT DETECTED
      CALL CHECKSIG (1,K,J,RANGE,JSKIP)
      IF (JSKIP.EQ.0) GO TO 140
      IF THIS TARGET IS OUTSIDE THE DETECTION RANGE SKIP IT
      JSKIP=0
      IF ((RANGE*1000).LE.RMAX1) JSKIP=1
      IF (JSKIP.EQ.0) GO TO 140
      TARGET IS WITHIN DETECTION RANGE
      EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE

```

```

C MASKING AND OBSCURANTS AND DROP THIS
C TARGET IF UNOBSERVABLE
C TERRAIN MASKING
C IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
C IF (JSKIP.EQ.0) GO TO 140
C VEHICLE MASKING
C CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C BURNING VEHICLE MASKING
C CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)
C IF (JSKIP.EQ.0) GO TO 140
C DUST MASKING
C IF THIS BLUE SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
C JDUST=1
C IF (ITRK1.EQ.0)
C & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE THROUGH IT
C SKIP THIS TARGET
C IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 140
C ENGINE SMOKE MASKING
C IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C JENG=1
C IF (ITRK2.EQ.0)
C & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
C IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 140
C SMOKE GRENADE CURTAIN MASKING
C IF THIS BLUE SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
C JSMK=1
C IF (ITRK3.EQ.0)
C & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
C IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 140
C ARTILLERY SMOKE CURTAIN MASKING
C IF THIS BLUE SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C ACQUISITION,
C EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS TARGET
C JART=1
C IF (ITRK4.EQ.0)
C & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
C IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 140
C THIS RED VEHICLE CAN BE TARGETED BY BLUE ARTILLERY
C ARTGTB(J,2)=1
140 CONTINUE
C END OF FINDING A TARGET FOR THIS BLUE SYSTEM
150 CONTINUE
ITLB=1
ENDIF
C LIST AVAILABLE TARGETS
WRITE (1,*) RED # SYS # SPEED HEADING X Y
KK=0
DO 20 K=1,200
  FORMAT (1X,I5,2X,I5,2X,I5,4X,F8.4,F8.4)
11  COUNTER-BATTERY FLAG (DISPLAY UNIT ID IF ARTILLERY AND SHOOTING)
  KFLAG=10
  IF (IRSYS(K,2).GT.0) KFLAG=IRAVL(K)
  IF (ARTGTB(K,2).EQ.1.OR.KFLAG.EQ.0) THEN
    WRITE (1,1) K,IRSYS(K,1),ISPDR(K),IRSA(K,2),RXA(K,1),RYA(K,2)
    KK=KK+1
    ENDF
  IF (ARTGTB(K,1).EQ.1.AND.ARTGTB(K,2).EQ.0.AND.KFLAG.NE.0) THEN
12  FORMAT (1X,I5,2X,I5,2X,I5,4X,F8.4,F8.4)
    WRITE (1,12) K,ISPDR(K),IRSA(K,2),RXA(K,1),RYA(K,2)
    KK=KK+1
    ENDF
  IF (KK.EQ.20) THEN
    READ (1,*)
    KK=0
  ENDF
20 CONTINUE
ELSE
  RED
  WRITE (1,*)
  WRITE (1,*) BLUE UNITS TARGETABLE FOR RED ARTILLERY FIRE
  WRITE (1,*)
  IF (ITLR.EQ.0) THEN
    DO 110 I=1,NB
      ARTGTR(I,1)=0
      ARTGTR(I,2)=0
      IF (IBSYS(I,2).GE.1) ARTGTR(I,1)=1
110 CONTINUE
    DO 1150 K=1,NR
      LOOP 1
      CHECK TO SEE IF THIS RED UNIT IS KILLED AND SKIP

```

```

C      IF (IRSYS(K,2).EQ.0.OR.IRSYS(K,1).GT.10) GO TO 1150
CCCCC  FIND ALL BLUE TARGETS          LOOP 3
C      FIND ANGLE AND RANGE TO ALL FRIENDLY AND THREAT VEHICLES
C      SO THAT MASKING CAN BE EVALUATED
C      KSIDE=2
C      MYSIDE=2
C      IRED=K
C      CALL VMASK (KSID, IRED)
C      FIND ALL BLUE TARGETS OUT TO 6000 METERS RANGE
C      NSYS=IRSYS(K,1)
C      RMAX1=6000
C      GET X,Y COORDINATES FOR THIS RED UNIT
C      XB=RXVA(K,1)
C      YB=RYVA(K,2)
C      IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C      FIND ANGLE AND RANGE TO ALL DUST PUFFS SO THAT MASKING CAN BE
C      EVALUATED
C      ITRK1=ITRK1(NSYS,1)
C      IF (ITRK1.EQ.0)
C      & CALL DMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C      FIND ANGLE AND RANGE TO ALL BURNING VEHICLES, SINCE THEY COMPLETELY
C      MASK VISUAL AND IR SYSTEMS
C      CALL COOKMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C      IF THIS RED SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C      FIND ANGLE AND RANGE TO ALL ENGINE SMOKE PUFFS SO THAT MASKING
C      CAN BE EVALUATED
C      ITRK2=ITRK2(NSYS,2)
C      IF (ITRK2.EQ.0)
C      & CALL ESMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C      IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C      FIND ANGLE AND RANGE TO ALL SMOKE GRENADE CURTAINS SO THAT MASKING
C      CAN BE EVALUATED
C      ITRK3=ITRK3(NSYS,3)
C      IF (ITRK3.EQ.0)
C      & CALL GRMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C      IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C      ACQUISITION,
C      FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE CURTAINS SO THAT
C      MASKING
C      CAN BE EVALUATED
C      ITRK4=ITRK4(NSYS,4)
C      IF (ITRK4.EQ.0)
C      & CALL ASMASK (XB,YB,RMAX1,RMAX1,RMAX1)
C      DO 1149 J=1,NB
C      LOOP 4
C      CHECK TO SEE IF THIS BLUE UNIT IS KILLED OR ALREADY IDENTIFIED
C      IF (IBSYS(J,2).EQ.0.OR.ARTGTR(J,2).EQ.1) GO TO 1149
C      FIND THIS BLUE UNIT'S X,Y COORDINATES
C      XR=BXVA(J,1)
C      YR=BYVA(J,2)
C      FIND RANGE TO THIS TARGET
C      RANGE=((XR-XB)**2+(YR-YB)**2)**.5
C      CHECK TARGET SIGNATURE AGAINST DETECTABILITY AND SKIP IF
C      NOT DETECTED
C      CALL CHECKSIG (2,K,J,RANGE,JSKIP)
C      IF (JSKIP.EQ.0) GO TO 1149
C      IF THIS TARGET IS OUTSIDE THE DETECTION RANGE SKIP IT
C      JSKIP=0
C      IF ((RANGE*1000.)>RMAX1) JSKIP=1
C      IF (JSKIP.EQ.0) GO TO 1149
C      TARGET IS WITHIN DETECTION RANGE
C      EVALUATE TERRAIN AND FRIENDLY/ENEMY VEHICLE
C      MASKING AND OBSCURANTS AND DROP THIS
C      TARGET IF UNOBSERVABLE
C      TERRAIN MASKING
C      IF (ITER.EQ.1) CALL TERRMASK (XB,YB,XR,YR,JSKIP)
C      IF (JSKIP.EQ.0) GO TO 1149
C      VEHICLE MASKING
C      CALL VSEE (MYSIDE,K,J,RANGE,AZIN,JSKIP,NB,NR)
C      BURNING VEHICLE MASKING
C      CALL COOKSEE (XB,YB,XR,YR,AZIN,JSKIP)
C      IF (JSKIP.EQ.0) GO TO 1149
C      DUST MASKING
C      IF THIS RED SYSTEM IS DUST SENSITIVE IN TARGET ACQUISITION,
C      EVALUATE ALL DUST PUFFS WITHIN RANGE OF THIS TARGET
C      JDUST=1
C      IF (ITRK1.EQ.0)
C      & CALL DSEE (XB,YB,XR,YR,AZIN,JDUST)
C      IF THERE IS DUST IN THE WAY AND BLUE CANNOT SEE THROUGH IT
C      SKIP THIS TARGET
C      IF (JDUST.EQ.0.AND.ITRK1.EQ.0) GO TO 1149
C

```

```

C      ENGINE SMOKE MASKING
C      IF THIS BLUE SYSTEM IS ENGINE SMOKE SENSITIVE IN TARGET ACQUISITION,
C      EVALUATE ALL ENGINE SMOKE PUFFS WITHIN RANGE OF THIS TARGET
C      JENG=1
C      IF (ITRK2.EQ.0)
C      & CALL ESSEE (XB,YB,XR,YR,AZIN,JENG)
C      IF (JENG.EQ.0.AND.ITRK2.EQ.0) GO TO 1149
C      SMOKE GRENADE CURTAIN MASKING
C      IF THIS RED SYSTEM IS SMOKE GRENADE SENSITIVE IN TARGET ACQUISITION,
C      EVALUATE ALL SMOKE GRENADE CURTAINS WITHIN RANGE OF THIS TARGET
C      JSMK=1
C      IF (ITRK3.EQ.0)
C      & CALL GRSEE (XB,YB,XR,YR,AZIN,JSMK)
C      IF (JSMK.EQ.0.AND.ITRK3.EQ.0) GO TO 1149
C      ARTILLERY SMOKE CURTAIN MASKING
C      IF THIS RED SYSTEM IS ARTILLERY SMOKE SENSITIVE IN TARGET
C      ACQUISITION,
C      EVALUATE ALL ARTILLERY SMOKE CURTAINS WITHIN RANGE OF THIS TARGET
C      JART=1
C      IF (ITRK4.EQ.0)
C      & CALL ASSEE (XB,YB,XR,YR,AZIN,JART)
C      IF (JART.EQ.0.AND.ITRK4.EQ.0) GO TO 1149
C      THIS BLUE VEHICLE CAN BE TARGETED BY RED ARTILLERY
C      ARTGTR(J,2)=1
1149  CONTINUE
C      END OF FINDING A TARGET FOR THIS RED SYSTEM
1150  CONTINUE
C      ICLR=1
C      ENDF
C      LIST AVAILABLE TARGETS
C      WRITE (,") 'BLUE# SYS# SPEED HEADING X Y'
C      KK=0
C      DO 120 K=1,200
C      KFLAG=10
C      IF (IBSYS(K,2).GT.0) KFLAG=IBAVL(K)
C      IF (ARTGTR(K,2).EQ.1.OR.KFLAG.EQ.0) THEN
C      WRITE (,"11) K,IBSYS(K,1),ISPD8(K),IBSA(K,2),BXVA(K,1),BYVA(K,2)
C      KK=KK+1
C      ENDF
C      IF (ARTGTR(K,1).EQ.1.AND.ARTGTR(K,2).EQ.0.AND.KFLAG.NE.0) THEN
C      WRITE (,"12) K,ISPD8(K),IBSA(K,2),BXVA(K,1),BYVA(K,2)
C      KK=KK+1
C      ENDF
C      IF (KK.EQ.20) THEN
C      READ (,")
C      KK=0
C      ENDF
120  CONTINUE
C      ENDF
C      RETURN
C      END
SUBROUTINE ARTRNG (KSID)
C      DISPLAYS CURRENTLY AVAILABLE ARTILLERY UNITS WITHIN RANGE
C      OF A SELECTED TARGET
COMMON/AAVAIL/IBAVL(200),IRAVL(200),NRARTB(200,5),NRARTR(200,5)
COMMON/ARTGTG/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR
COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
COMMON/MOVE/BXYA(200,2),RXVA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
COMMON/MANEUVER/BXYO(200,2),RXYO(200,2),MTYPEB(200),MTYPEP(200)
& BXYP(200,2),RXYP(200,2),ISPD8(200),ISPD8(200)
COMMON/TRACKING/BOREB(10,3),BOREP(10,3),IRNGB(30,4),IRNGP(30,4),
& ITRKB(10,4),ITRKP(10,4),TOFB(10,3),TOFP(10,3)
COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
& XYFUR(200,2),MILAB(200,10),MILAR(200,10)
COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
& NAMMOB(10,5),NAMMOP(10,5),IATYPB(10,5),IATYPP(10,5)
& IAFAPB(10,5),IAFAFP(10,5),IARNGB(10,5,5),IARNGP(10,5,5),
& ARTPKB(20,10,5),ARTPKP(20,10,5),IABTUBE(10),IARTUBE(10),
& ISCOOTB(10,4),ISCOOTP(10,4)
INTEGER ARTGTB,ARTGTR
DIMENSION RMAX(5),IRMAX(5)
CHARACTER*9 BANAME,RANAME
CHARACTER*16 BHIER,RHIER,IBIER
CHARACTER*24 BUNIT,RUNIT
CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
C      AVAILABILITY FLAGS
C      1 -- AVAILABLE FOR A FIRE MISSION
C      IF (KSID.EQ.1) THEN

```







```

MLTYPE=IAFAFB(KASYS,ITYPE)
IF (MLTYPE.EQ.1.AND.ARTGTB(NTARG,2).EQ.0) THEN
  WRITE (":") 'TARGET "NTARG" IS NOT TARGETABLE WITH A M-L
&PROJECTILE, RESELECT PROJECTILE TYPE'
  ITYPE=0
  WRITE (":") 'PAUSE ... ENTER TO CONTINUE'
  READ (":")
  GO TO 14
ENDIF
IRMAX=IARNGB(KASYS,ITYPE,1)
ITOFMAX=IARNGB(KASYS,ITYPE,2)

C   WRITE (":") 'IRMAX, ITOFMAX'
C   WRITE (":") 'IRMAX, ITOFMAX'

  IF (IATYPB(KASYS,ITYPE).EQ.3) NTARG=-1

  ENDF
ENDIF
15 IF (NUMR.GT.0.AND.ITYPE.NE.0.AND.NUMA.NE.0) THEN
  MAXRND=NRARTB(NUMA,ITYPE)
  IF (NUMR.GT.MAXRND) THEN
    WRITE (":") 'INSUFFICIENT AMMO TO FILL REQUEST'
    WRITE (":") 'MAX AVAILABLE OF THIS TYPE "MAXRND'
    WRITE (":") 'RESELECT NUMBER OF ROUNDS TO FIRE'
    NUMR=0
    WRITE (":") 'PAUSE ... ENTER TO CONTINUE'
    READ (":")
    GO TO 15
  ENDF
  ISCOOT=ISCOOTB(KASYS,1)
  MAXSCOOT=ISCOOTB(KASYS,2)
  IF (NUMR.GT.MAXSCOOT.AND.ISCOOT.EQ.1) THEN
    WRITE (":") 'THIS SYSTEM SHOTS AND SCOOT'
    WRITE (":") 'MAX # ROUNDS EACH MISSION "MAXSCOOT'
    WRITE (":") 'RESELECT NUMBER OF ROUNDS TO FIRE'
    NUMR=0
    WRITE (":") 'PAUSE ... ENTER TO CONTINUE'
    READ (":")
    GO TO 15
  ENDF
  ELSE
    NUMR=0
  ENDF
16 IF (IDELAY.NE.0) THEN
  IF (IDELAY.LT.ITSTART) THEN
    WRITE (":") 'DELAY FIRING TIME IS LESS THAN MINIMUM'
    WRITE (":") 'SETUP TIME ... DELAY TIME SET TO 0'
    WRITE (":")
    IDELAY=0
    WRITE (":") 'PAUSE ... ENTER TO CONTINUE'
    READ (":")
  ENDF
  ENDF
ENDIF

C   LOOP 2
C   IF (NTARG.NE.0.AND.NUMA.NE.0.AND.ITYPE.NE.0) THEN
C   CALCULATE ESTIMATED TIME OF ARRIVAL AND FINAL RANGE TO
C   THE TARGET
C   IF TARGET IS MOVING, SOLVE QUADRATIC FORMULA FOR ESTIMATED
C   POSITION AND ETA
  IF (ISPD.EQ.0) THEN
    &"**2**5)
    IRANGE=NINT((REAL(XT)-REAL(XA))**2+(REAL(YT)-REAL(YA))
    IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
    IF (IETA.GT.ITOFMAX) THEN
      WRITE (":") 'TARGET "NTARG" IS OUT OF RANGE OF THIS ARTILLERY
&UNIT/AMMUNITION PAIR - SELECT AGAIN'
      IRANGE=0
      IETA=0
      GO TO 10
    ELSE
      IETA=IETA+ITSTART
      IF (IDELAY.GT.ITSTART) IETA=IETA+(IDELAY-ITSTART)
    ENDF
  ELSE
    XO=REAL(XT)
    YO=REAL(YT)
    XA=REAL(XA)
    YA=REAL(YA)
    V=REAL(ISPD)*1000/3600
    CK=REAL(ITOFMAX)/REAL(IRMAX)
    COST=COS(REAL(IDIR/180.*PI))
    SINT=SIN(REAL(IDIR/180.*PI))
    ITGO=ITSTART
    IF (IDELAY.GT.ITSTART) ITGO=ITGO+(IDELAY-ITSTART)

C   WRITE (":") 'ITGO'
C   WRITE (":") 'ITGO'

    XO=XO-V*COST*REAL(ITGO)
    YO=YO-V*SINT*REAL(ITGO)

C   WRITE (":") 'XO,YO,XA,YA'
C   WRITE (":") 'XO,YO,XA,YA'

    A=V*V*COST*ABS(COST)*CK*CK-V*V*SINT*ABS(SINT)*CK*CK-1
    B=2*(XO-XA)*V*COST*CK+2*(YO-YA)*V*SINT*CK
    C=(XO-XA)**2+(YO-YA)**2

C   WRITE (":") 'A,B,C'
C   WRITE (":") 'A,B,C'

R1=(-B+(B*B-4.*A*C)**.5)/2.*A
R2=(-B-(B*B-4.*A*C)**.5)/2.*A

C   WRITE (":") 'R1,R2'
C   WRITE (":") 'R1,R2'

  IF (R1.GE.R2) RANGE=R1
  IF (R2.GT.R1) RANGE=R2
  IRANGE=NINT(RANGE)
  IETA=NINT((REAL(RANGE)/REAL(IRMAX))*REAL(ITOFMAX))
  IF (IETA.GT.ITOFMAX) THEN
    WRITE (":") 'TARGET "NTARG" IS MOVING OUT OF RANGE OF THIS
*ARTILLERY UNIT/AMMUNITION PAIR - SELECT AGAIN'
    IRANGE=0
    IETA=0
    GO TO 10
  ELSE
    IXT=NINT(XO-V*COST*REAL(IETA))
    IYT=NINT(YO-V*SINT*REAL(IETA))
    IETA=IETA+ITGO
  ENDF
  ENDF
ENDIF
END LOOP 2

C   ENDIF
C   GO TO 10

100 CONTINUE

C   CHECK TO SEE THAT ALL REQUIRED MISSION DATA FIELDS ARE FILLED
C   AND THEN ENTER MISSION INTO BACKLOG

  IF (NUMA.EQ.0.OR.NTARG.EQ.0.OR.NUMR.EQ.0.OR.ITYPE.EQ.0) THEN
101 WRITE (":") 'MISSION FIELDS INCOMPLETE'
    WRITE (":") '1 - RETURN TO INPUT MENU'
    WRITE (":") '0 - CANCEL FIRE MISSION'
    WRITE (":")
    READ (":",ERR=101) ISTOP
    IF (ISTOP.EQ.1) GO TO 10
    IF (ISTOP.EQ.0) GO TO 200
    GO TO 101
  ENDF

102 IF (IAFAFB(KASYS,ITYPE).EQ.1.AND.IGUIDE(1).EQ.0) THEN
  WRITE (":") 'NO TARGET IN GUIDED PROJECTILE TARGET LIST'
  WRITE (":") '1 - RETURN TO INPUT MENU'
  WRITE (":") '0 - CANCEL FIRE MISSION'
  WRITE (":")
  READ (":",ERR=102) ISTOP
  IF (ISTOP.EQ.1) GO TO 10
  IF (ISTOP.EQ.0) GO TO 200
  GO TO 102
  ENDF

C   ENTER DATA INTO FIRE MISSION LOG
  K=NUMNEW
  NFMIS(1)=K
  MFUB(K,1)=NUMA
  MFUB(K,2)=NUMR
  MFUB(K,3)=ITYPE
  MFUB(K,4)=IAREA
  MFUB(K,5)=IETA
  XYFUB(K,1)=REAL(IXT)/1000
  XYFUB(K,2)=REAL(IYT)/1000
  DO 103 IK=1,10
103 MILAB(K,IK)=IGUIDE(IK)
  ISSA(NUMA,1)=0
  ISPBS(NUMA)=0
  IBAVL(NUMA)=0

C   SET UP SHOOT AND SCOOT INFO AND RE-AVAILABILITY TIME FOR
C   THIS BLUE ARTILLERY UNIT
  IF (ISCOOT.EQ.1) THEN
    ITSC(NUMA,1)=ITGO+80+ITIME
    ITAVL(NUMA,1)=ITGO+80+ISCOOTB(KASYS,4)+ITIME
  ELSE
    ITSC(NUMA,1)=0
    ITAVL(NUMA,1)=ITGO+80+ISCOOTB(KASYS,4)+ITIME
  ENDF

C   EXPEND APPROPRIATE NUMBER OF ROUNDS PER TUBE
  NRARTB(NUMA,ITYPE)=NRARTB(NUMA,ITYPE)-NUMR
  GO TO 5

C   ELSE
C   (ELSE) LOOP 1

C   RED
  NUMNEW=NFMIS(2)+1
  WRITE (":") 'RED'
  WRITE (":") 'NEW MISSION NUMBER "NUMNEW'
  WRITE (":") '1 - ARTILLERY UNIT # "NUMA'
  WRITE (":") '2 - TARGET UNIT # "NTARG'
  WRITE (":") 'PROJECTED TARGET POSITION: XT (M) "XT, YT (M) "YT'
  WRITE (":") '3 - BEATEN AREA (M**2) OR BASKET AREA "IAREA'
  WRITE (":") '4 - # ROUNDS PER TUBE "NUMR'
  WRITE (":") '5 - ROUND TYPE # "ITYPE'
  WRITE (":") '8 - DELAY FIRING TIME "IDELAY'
  WRITE (":") 'ETA (sec) - "IETA'
  WRITE (":") 'TARGET RANGE - "IRANGE'
  WRITE (":")

```

```

IF (MILTYPE.EQ.1) THEN
  WRITE (',') 7 - GUIDED PROJECTILE TARGET LIST
  WRITE (',') (IGUIDE(J),J=1,10)
  WRITE (',')
ENDIF
IF (NTARG.EQ.-1) THEN
  WRITE (',') 8 - SMOKE ROUNDS AIMPOINT XT,YT
  WRITE (',') IXT,IYT
  WRITE (',')
ENDIF
IF (NUMA.GT.0.AND.NTARG.GT.0.AND.NUMR.GT.0.AND.ITYPE.GT.0) THEN
  KSYS=IRSYS(NUMA,1)
  IF (KSYS.GT.10) THEN
    KASY=KSYS-10
    IFC=IATYPR(KASY,ITYPE)
    IF (IFC.EQ.2) THEN
C ESTIMATE AREA TARGET PATTERN EFFECTIVENESS
      CALL FCEST (2,NUMA,KASY,ITYPE,NUMR,NTARG,AREA)
      ENDF
    ENDF
  ENDIF
  WRITE (',')
  WRITE (',') 'ENTER ITEM NUMBER TO CHANGE (ONE AT A TIME)'
  WRITE (',') '0 - QUIT'
  WRITE (',')
  READ (',',ERR=10) ITEM
  IF (ITEM.LT.0.OR.ITEM.GT.8) GO TO 10
  IF (ITEM.EQ.0) GO TO 3100
311 IF (ITEM.EQ.7) THEN
    WRITE (',') 'ENTER UP TO TEN TARGET NUMBERS (FILL WITH 0)'
    WRITE (',')
    READ (',',ERR=311) (IGUIDE(J),J=1,10)
  ELSE
309 IF (ITEM.EQ.8) THEN
    WRITE (',') 'ENTER XT, YT SMOKE AIMPOINT (METERS)'
    WRITE (',')
    READ (',',ERR=309) IXT,IYT
  ELSE
    WRITE (',') 'ENTER NEW VALUE OF ITEM ',ITEM
    WRITE (',')
    READ (',',ERR=311) NITEM
  ENDIF
C IF (NITEM.LT.0) GO TO 311
  ENDF
  IF (ITEM.EQ.1) NUMA=NITEM
  IF (ITEM.EQ.2) NTARG=NITEM
  IF (ITEM.EQ.3) IAREA=NITEM
  IF (ITEM.EQ.4) NUMR=NITEM
  IF (ITEM.EQ.5) ITYPE=NITEM
  IF (ITEM.EQ.6) IDLAY=NITEM
312 IF (NTARG.EQ.0) THEN
    IXT=0
    IYT=0
    IRANGE=0
    IETA=0
    IDIR=0
    IAZ=0
    ISPD=0
    ENDF
    IF (NTARG.EQ.-1) ISPD=0
    IF (ARTGTR(NTARG,1).EQ.0.AND.ARTGTR(NTARG,2).EQ.0) THEN
      WRITE (',') 'TARGET ',NTARG,' IS NOT ON TARGETABLE LIST'
      NTARG=0
      WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
      READ (',')
      GO TO 312
    ENDF
    IXT=NINT(BXYA(NTARG,1)*1000.)
    IYT=NINT(BXYA(NTARG,2)*1000.)
    ISPD=ISPD8(NTARG)
    IAZ=IBSA(NTARG,2)
    AZIN=REAL(IAZ)
    CALL FDIR(AZIN,AZOUT)
    IDIR=NINT(AZOUT)
C WRITE (',') IXT, IYT, ISPD, IAZ, IDIR
C WRITE (',') IXT,IYT,ISPD,IAZ,IDIR
  ENDF
313 IF (NUMA.EQ.0) THEN
    KSYS=0
    IXT=0
    IYT=0
    IRANGE=0
    IETA=0
    IMOVE=0
    ITSTART=15
  ELSE
    KSYS=IRSYS(NUMA,1)
    KASY=IRSYS(NUMA,1)-10
    IF (KSYS.LE.10) THEN
      WRITE (',') 'RED ',NUMA,' IS NOT AN ARTILLERY UNIT'
      NUMA=0
      WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
      READ (',')
      GO TO 313
    ENDF
    IF (IRAVL(NUMA,NE.1) THEN
      WRITE (',') 'RED ARTILLERY UNIT',NUMA,' IS NOT AVAILABLE'
      NUMA=0
      WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
    ENDIF
  ENDIF

```

```

  READ (',')
  GO TO 313
ENDIF
IXA=NINT(RXYA(NUMA,1)*1000.)
IYA=NINT(RXYA(NUMA,2)*1000.)
MOVE=ISPD8(NUMA)
C WRITE (',') IXA, IYA, MOVE
C WRITE (',') IXA,IYA,MOVE
  IF (IMOVE.GT.0) THEN
    ITSTART=60
  ELSE
    ITSTART=15
  ENDIF
ENDIF
314 IF (ITYPE.EQ.0) THEN
  IXT=0
  IYT=0
  IRANGE=0
  IETA=0
  MILTYPE=0
  IRMAX=0
  ITOFMAX=0
  ELSE
    IF (KSYS.NE.0) THEN
      MAXTYPE=0
      IF (NRARTR(NUMA,1).GT.0) MAXTYPE=1
      IF (NRARTR(NUMA,2).GT.0) MAXTYPE=2
      IF (NRARTR(NUMA,3).GT.0) MAXTYPE=3
      IF (NRARTR(NUMA,4).GT.0) MAXTYPE=4
      IF (NRARTR(NUMA,5).GT.0) MAXTYPE=5
      IF (ITYPE.GT.MAXTYPE) THEN
        WRITE (',') 'NO AMMO AVAILABLE FOR TYPE ',ITYPE
        ITYPE=0
        WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
        READ (',')
        GO TO 314
      ENDF
      KASY=IRSYS(NUMA,1)-10
      MILTYPE=IAFAFR(KASY,ITYPE)
      IF (MILTYPE.EQ.1.AND.ARTGTR(NTARG,2).EQ.0) THEN
        WRITE (',') 'TARGET ',NTARG,' IS NOT TARGETABLE WITH A M-I-L
        &PROJECTILE. RESELECT PROJECTILE TYPE'
        ITYPE=0
        WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
        READ (',')
        GO TO 314
      ENDF
      IRMAX=IARNGR(KASY,ITYPE,1)
      ITOFMAX=IARNGR(KASY,ITYPE,2)
      WRITE (',') 'IRMAX, ITOFMAX'
      WRITE (',') IRMAX,ITOFMAX
    C
    C IF (IATYPR(KASY,ITYPE).EQ.3) NTARG=-1
  ENDF
ENDIF
315 IF (NUMR.GT.0.AND.ITYPE.NE.0.AND.NUMA.NE.0) THEN
  MAXRND=NRARTR(NUMA,ITYPE)
  IF (NUMR.GT.MAXRND) THEN
    WRITE (',') 'INSUFFICIENT AMMO TO FILL REQUEST'
    WRITE (',') 'MAX AVAILABLE OF THIS TYPE ',MAXRND
    WRITE (',') 'RESELECT NUMBER OF ROUNDS TO FIRE'
    NUMR=0
    WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
    READ (',')
    GO TO 315
  ENDF
  ISCOOT=ISCOOTR(KASY,1)
  MAXSCOOT=ISCOOTR(KASY,2)
  IF (NUMR.GT.MAXSCOOT.AND.ISCOOT.EQ.1) THEN
    WRITE (',') 'THIS SYSTEM SHOOTS AND SCOOT'S'
    WRITE (',') 'MAX # ROUNDS EACH MISSION ',MAXSCOOT
    WRITE (',') 'RESELECT NUMBER OF ROUNDS TO FIRE'
    NUMR=0
    WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
    READ (',')
    GO TO 315
  ENDF
  ELSE
    NUMR=0
  ENDF
316 IF (IDLAY.NE.0) THEN
  IF (IDLAY.LT.ITSTART) THEN
    WRITE (',') 'DELAY FIRING TIME IS LESS THAN MINIMUM'
    WRITE (',') 'SETUP TIME ... DELAY TIME SET TO 0'
    WRITE (',')
    IDLAY=0
    WRITE (',') 'PAUSE ... ENTER TO CONTINUE'
    READ (',')
  ENDF
ENDIF
C
C LOOP 2
C IF (NTARG.NE.0.AND.NUMA.NE.0.AND.ITYPE.NE.0) THEN
C CALCULATE ESTIMATED TIME OF ARRIVAL AND FINAL RANGE TO
C THE TARGET
C IF TARGET IS MOVING, SOLVE QUADRATIC FORMULA FOR ESTIMATED
C POSITION AND ETA
C IF (ISPD.EQ.0) THEN
  IRANGE=NINT(((((REAL(IXT)-REAL(IXA))**2+(REAL(IYT)-REAL(IYA))

```

```

&"2)"5)
  IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
  IF (IETA.GT.ITOFMAX) THEN
    WRITE (" ") 'TARGET 'NTARG' IS OUT OF RANGE OF THIS ARTILLERY
    &UNIT/AMMUNITION PAIR - SELECT AGAIN
    IRANGE=0
    IETA=0
    GO TO 10
  ELSE
    IETA=IETA+ITSTART
    IF (IDELAY.GT.ITSTART) IETA=IETA+(IDELAY-ITSTART)
  ENDF
ELSE
  X0=REAL(IXT)
  Y0=REAL(IYT)
  XA=REAL(IXA)
  YA=REAL(IYA)
  V=REAL(ISPD)*1000/3600
  CK=REAL(ITOFMAX)/REAL(IRMAX)
  COST=COS(REAL(IDIRV/180.*PI))
  SINT=SIN(REAL(IDIRV/180.*PI))
  ITGO=ITSTART
  IF (IDELAY.GT.ITSTART) ITGO=ITGO+(IDELAY-ITSTART)
C   WRITE (" ") 'ITGO'
C   WRITE (" ") ITGO
  X0=X0-V*COST*REAL(ITGO)
  Y0=Y0-V*SINT*REAL(ITGO)
C   WRITE (" ") X0,Y0,XA,YA
C   WRITE (" ") X0,Y0,XA,YA
  A=V*V*COST*ABS(COST)*CK*CK+V*V*SINT*ABS(SINT)*CK*CK-1
  B=2*(X0-XA)*V*COST*CK+2*(Y0-YA)*V*SINT*CK
  C=(X0-XA)**2+(Y0-YA)**2
C   WRITE (" ") A,B,C
C   WRITE (" ") A,B,C
  R1=(-B+(B*B-4.*A*C)**.5)/(2.*A)
  R2=(-B-(B*B-4.*A*C)**.5)/(2.*A)
C   WRITE (" ") R1,R2
C   WRITE (" ") R1,R2
  IF (R1.GE.R2) RANGE=R1
  IF (R2.GT.R1) RANGE=R2
  IRANGE=NINT(RANGE)
  IETA=NINT((REAL(IRANGE)/REAL(IRMAX))*REAL(ITOFMAX))
  IF (IETA.GT.ITOFMAX) THEN
    WRITE (" ") 'TARGET 'NTARG' IS MOVING OUT OF RANGE OF THIS
    'ARTILLERY UNIT/AMMUNITION PAIR - SELECT AGAIN'
    IRANGE=0
    IETA=0
    GO TO 10
  ELSE
    IXT=NINT(X0+V*COST*REAL(IETA))
    IYT=NINT(Y0+V*SINT*REAL(IETA))
    IETA=IETA+ITGO
  ENDF
ENDIF
C   ENDIF
C   END LOOP 2
GO TO 10
3100 CONTINUE
C   CHECK TO SEE THAT ALL REQUIRED MISSION DATA FIELDS ARE FILLED
C   AND THEN ENTER MISSION INTO BACKLOG
  IF (NUMA.EQ.0.OR.NTARG.EQ.0.OR.NUMR.EQ.0.OR.ITYPE.EQ.0) THEN
3101  WRITE (" ") 'MISSION FIELDS INCOMPLETE'
    WRITE (" ") '1 - RETURN TO INPUT MENU'
    WRITE (" ") '0 - CANCEL FIRE MISSION'
    WRITE (" ")
    READ (" ",ERR=3101) ISTOP
    IF (ISTOP.EQ.1) GO TO 10
    IF (ISTOP.EQ.0) GO TO 200
    GO TO 3101
  ENDF
3102  IF (IAFAFR(KASYS,ITYPE).EQ.1.AND.IGUIDE(1).EQ.0) THEN
    WRITE (" ") 'NO TARGET IN GUIDED PROJECTILE TARGET LIST'
    WRITE (" ") '1 - RETURN TO INPUT MENU'
    WRITE (" ") '0 - CANCEL FIRE MISSION'
    WRITE (" ")
    READ (" ",ERR=3102) ISTOP
    IF (ISTOP.EQ.1) GO TO 10
    IF (ISTOP.EQ.0) GO TO 200
    GO TO 3102
  ENDF
C   ENTER DATA INTO FIRE MISSION LOG
  K=NUMNEW
  NFMS(2)=K
  MFUR(K,1)=NUMA
  MFUR(K,2)=NUMR
  MFUR(K,3)=ITYPE
  MFUR(K,4)=IAREA
  MFUR(K,5)=IETA
  XYFUR(K,1)=REAL(IXT)/1000.

```

```

  XYFUR(K,2)=REAL(IYT)/1000.
  DO 3103 K=1,10
3103  MILAR(K,0)=IGUIDE(K)
  IRSA(NUMA,1)=0
  ISPD(NUMA)=0
  IRAVL(NUMA)=0
C   SET UP SHOOT AND SCOOT INFO AND RE-AVAILABILITY TIME FOR
C   THIS RED ARTILLERY UNIT
  IF (ISCOOT.EQ.1) THEN
    ITSC(NUMA,2)=ITGO+80+ITIME
    ITAVL(NUMA,2)=ITGO+80+ISCOOTR(KASYS,4)+ITIME
  ELSE
    ITSC(NUMA,2)=0
    ITAVL(NUMA,2)=ITGO+80+ISCOOTR(KASYS,4)+ITIME
  ENDF
C   EXPEND APPROPRIATE NUMBER OF ROUNDS PER TUBE
  NRARTR(NUMA,ITYPE)=NRARTR(NUMA,ITYPE)-NUMR
  GO TO 5
C   ENDIF
C   END LOOP 1
200 CONTINUE
RETURN
END
SUBROUTINE FIREMISH (ITIME,ITIME)
C   C
C   C
  RESOLVES CURRENT BACKLOG OF ARTILLERY IN-DIRECT FIRE MISSIONS
  COMMON/MOVE/BXYA(200,2),RXYA(200,2),ISSA(200,2),IRSA(200,2),
  & L1BM,L2M,L3M,L1RM
  COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
  & IBSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
  COMMON/ARTGT/ARTGTB(200,2),ARTGTR(200,2),ITLB,ITLR
  COMMON/AMISH/NFMIS(2),MFUB(200,5),MFUR(200,5),XYFUB(200,2),
  & XYFUR(200,2),MILAB(200,10),MILAR(200,10)
  COMMON/ARTYDATAB/ANRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10),
  & NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5),
  & IAFAB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
  & ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
  & ISCOOTB(10,4),ISCOOTR(10,4)
  COMMON/AAVAL/IBAVL(200),IRAVL(200),NPARTB(200,5),NPARTR(200,5)
  COMMON/SCOOT/ITSC(200,2),ITAVL(200,2)
  DIMENSION IGUIDE(10),ITARGR(200),ITARGB(200)
  CHARACTER*50 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
  CHARACTER*5 BNAME,RNAME,BANAME,RANAME
  CHARACTER*10 BHIER,RHIER,BIER
  CHARACTER*24 BUNIT,RUNIT
  INTEGER ARTGTB,ARTGTR
  PI=3.14159
  DO 10 IT=1,200
  ITARGB(IT)=0
  ITARGR(IT)=0
10
C   RESOLVE BLUE FIRE MISSIONS
  DO 500 N=1,NFMIS(1)
  IF (MFUB(N,1).EQ.0) GO TO 500
  IETA=MFUB(N,5)
  IF (IETA.GT.0) THEN
C   NOT ARRIVED ON TARGET YET
    MFUB(N,5)=MFUB(N,5)+ITIME
    GO TO 500
  ELSE
C   THIS MISSION HAS COME DUE
    NUMA=MFUB(N,1)
    XA=BXYA(NUMA,1)*1000.
    YA=BXYA(NUMA,2)*1000.
    KASYS=IBSYS(NUMA,1)+10
    NTUBES=IBSYS(NUMA,2)
    NUMR=MFUB(N,2)
    ITYPE=MFUB(N,3)
    MILTYPE=0
    IF (IAFAFR(KASYS,ITYPE).EQ.1) MILTYPE=1
    IATYPE=IATYPB(KASYS,ITYPE)
    IF (MILTYPE.EQ.1) IATYPE=0
    MAXRNG=IARNGB(KASYS,ITYPE,1)
    MAXCEP=IARNGB(KASYS,ITYPE,3)
    NSUBS=IARNGB(KASYS,ITYPE,4)
    IF (NSUBS.LE.0) NSUBS=1
    PATAREA=REAL(IARNGB(KASYS,ITYPE,5))
    BEATAREA=REAL(MFUB(N,4))
    XT=XYFUB(N,1)*1000.
    YT=XYFUB(N,2)*1000.
C   ADJUST IMPACT POINT BASED ON CEP AND RANDOM NUMBER
    RANGE=((XT-XA)**2+(YT-YA)**2)**.5
    CEPM=REAL(MAXCEP)*RANGE/REAL(MAXRNG)
    RDEP=.573*CEPM
    SIGRD=RDEP/.8745

```

```

CALL CEP (SIGMA)
SIGRO=SIGRO+SIGMA
XT=XT+SIGRO
YT=YT+SIGRO
ACEP=P1+SIGRO+SIGRO
IF (BEATAREA.LT.ACEP) BEATAREA=ACEP
IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
RADIUS=(BEATAREA/P1)**.5

C WRITE (".") ACEP, PATAREA, BEATAREA, RADIUS
C WRITE (".") ACEP, PATAREA, BEATAREA, RADIUS

DO 103 IK=1,10
103 IGUIDE(IK)=MILAS(N,IK)

IF (IATYPE.EQ.3) THEN
CCC CREATE ARTILLERY SMOKE CLOUD
CCC
RADIUS=ARTPKB(1,KASYS,ITYPE)
A1=PTRAD*SRADIUS
AT=A1*NUMA*RTUBES
IF (BEATAREA.GE.AT) THEN
RADIUS=(AT/P1)**.5
ELSE
RADIUS=(BEATAREA/P1)**.5
ENDIF
TSMK=ARTPKB(2,KASYS,ITYPE)
CALL ARTSMOKE (ITIME,XT,YT,RADIUS,TSMK)
ENDIF

IF (MILTYPE.EQ.1) THEN
C EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C TARGET LIST
C LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

C FIND ALL RED TARGETS IN BASKET AREA
DO 110 IT=1,NR
IF (IRSYS(IT,2).LE.0) GO TO 110
XR=RXA(IT,1)*1000.
YR=RYA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS RED VEHICLE CAN BE TARGETED
ITARGR(IT)=1
ENDIF
110 CONTINUE

C COMPARE TARGET LIST AGAINST AVAILABLE TARGETS
NSHOTS=NUMR*NTUBES*NSUBS

115 DO 120 IT=1,10
C IF NO SHOTS LEFT, SKIP IT
IF (NSHOTS.LE.0) GO TO 120
NT=IGUIDE(IT)
C IF NO TARGET ON LIST, SKIP IT
IF (NT.EQ.0) GO TO 120
C IF ALREADY KILLED SKIP IT
IF (IRSYS(NT,2).LE.0) THEN
IGUIDE(IT)=0
GO TO 120
ENDIF
C SEE IF THIS TARGET IS IN BASKET AND THEN SHOOT IT
IF (ITARGR(IT).EQ.0) THEN
IGUIDE(IT)=0
GO TO 120
ENDIF
C SHOOT IT
JSYS=IRSYS(NT,1)
PKIT=ARTPKB(JSYS,KASYS,ITYPE)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IRSYS(NT,2)
JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NKILLS=0
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
&MILTYPE,IATYPE)
DO 118 INK=1,NKILLS
WRITE (10,") RED 'NT' KILLED BY BLUE ARTILLERY 'NUMA,
& 'M-I-L AMMO TYPE',ITYPE
CALL WARTY (ITIME,2,NT,1,NUMA,ITYPE)
CALL OUTPUT2 (ITIME,2,NT,0)
118 CONTINUE
IRSYS(NT,2)=IRSYS(NT,2)-NKILLS
IGUIDE(IT)=0
ITARGR(NT)=0
ELSE
PROB=RND0
IF (PROB.LE.PKIT) THEN
C TARGET KILLED
WRITE (10,") RED 'NT' KILLED BY BLUE ARTILLERY 'NUMA,
& 'M-I-L AMMO TYPE',ITYPE
IRSYS(NT,2)=IRSYS(NT,2)-1
IF (IRSYS(NT,2).LT.0) IRSYS(NT,2)=0
C REMOVE FROM TARGET LIST, SINCE KILLED
IGUIDE(IT)=0
ITARGR(NT)=0
CALL WARTY (ITIME,2,NT,1,NUMA,ITYPE)

```

```

CALL OUTPUT2 (ITIME,2,NT,0)
ENDIF
NSHOTS=NSHOTS-1
ENDIF
120 CONTINUE

NUM=0
C IF ANY SHOTS REMAIN, GO DOWN TARGETABLE LIST AND SHOOT AT THEM
IF (NSHOTS.GE.1) THEN
DO 130 IT=1,NR
IF (NUM.EQ.10) GO TO 130
IF (NUM.EQ.NSHOTS) GO TO 130
IF (ITARGR(IT).GT.0) THEN
NUM=NUM+1
IGUIDE(NUM)=IT
ENDIF
130 CONTINUE
ENDIF
IF (NUM.GT.0) GO TO 115
ENDIF

IF (IATYPE.EQ.1) THEN
C EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C ROUNDS
C FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C PROBABILITIES OF BEING TARGETED AND KILLED
C REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISNT, FRIENDLY
C VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.

NTR=0
C FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
DO 210 IT=1,NR
XR=RXA(IT,1)*1000.
YR=RYA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS RED VEHICLE CAN BE TARGETED
ITARGR(IT)=1
NTR=NTR+1
ENDIF
210 CONTINUE

NTB=0
DO 211 IT=1,NR
XR=RXA(IT,1)*1000.
YR=RYA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS BLUE VEHICLE CAN BE TARGETED
ITARGB(IT)=1
NTB=NTB+1
ENDIF
211 CONTINUE
NTOTAL=NTR+NTB
IF (NTOTAL.LE.0) GO TO 300

C FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
NSHOTS=NUMR*NTUBES*NSUBS
SEACH=REAL(NSHOTS)/REAL(NTOTAL)
C BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C THAT A VEHICLE WILL BE WITHIN RANGE OF AT LEAST ONE SUBMUNITION
P1=1-EXP(-SEACH)

C LIMIT SHOOT OUT TO NSUBS CYCLES, SINCE THE UNIFORMITY OF
C THE PATTERN AND TARGET DENSITIES ARE UNKNOWN.
C IT IS REASONABLE TO ASSUME THAT ANY ONE TARGET WILL BE
C WITHIN THE ATTACK RANGE OF ANY ONE MISSILE.

NCYC=0
215 IF (NCYC.GT.NSUBS) GO TO 300

DO 220 IR=1,NR
IF (NSHOTS.LE.0) GO TO 300
IF (ITARGR(IR).NE.1) GO TO 220
PROB=RND0
IF (PROB.GT.P1) GO TO 220
C SHOOT IT
JSYS=IRSYS(IR,1)
PKIT=ARTPKB(JSYS,KASYS,ITYPE)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IRSYS(IR,2)
JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NKILLS=0
NSHOT=NSUBS

C WRITE (".") CALL COUNTERB, NKILLS,JTUBES,DTUBES,RADIUS,
C &NSHOT,PKIT,MILTYPE,IATYPE
C WRITE (".") NKILLS,JTUBES,DTUBES,RADIUS,
C &NSHOT,PKIT,MILTYPE,IATYPE

CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,PKIT,

```

```

&MILTYPE,IATYPE)
C WRITE (',') 'NKILLS'
C WRITE (',') 'NKILLS'

    NSHOTS=NSHOTS-(NSUBS-NSHOT)
    DO 216 INK=1,NKILLS
      WRITE (10,') 'RED',IR,' KILLED BY BLUE ARTILLERY',NUMA,
        & ' F-A-F GUIDED AMMO TYPE',ITYPE
      CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
      CALL OUTPUT2 (ITIME,2,IR,0)
216 CONTINUE
      IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
      IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
      ELSE
        PROB=RND0
        IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
          C TARGET KILLED
          & WRITE (10,') 'RED',IR,' KILLED BY BLUE ARTILLERY',NUMA,
            & ' F-A-F GUIDED AMMO TYPE',ITYPE
          IRSYS(IR,2)=IRSYS(IR,2)-1
          IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
          CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
          CALL OUTPUT2 (ITIME,2,IR,0)
          ENDF
          NSHOTS=NSHOTS-1
        ENDF
220 CONTINUE
        DO 221 IB=1,NB
          IF (NSHOTS.LE.0) GO TO 300
          IF (ITARGB(IB).NE.1) GO TO 221
          PROB=RND0
          IF (PROB.GT.P1) GO TO 221
          C SHOOT IT
          JSYS=IBSYS(IB,1)
          PKIT=ARTPKB(JSYS,KASYS,ITYPE)
          IF (JSYS.GT.10) THEN
            C COUNTER-BATTERY SITUATION
            JTUBES=IBSYS(IB,2)
            JASYS=JSYS-10
            DTUBES=REAL(IABTUBE(JASYS))
            NKILLS=0
            NSHOT=NSUBS
            CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,PKIT,
              &MILTYPE,IATYPE)
            NSHOTS=NSHOTS-(NSUBS-NSHOT)
            DO 217 INK=1,NKILLS
              WRITE (10,') 'BLUE',IB,' KILLED BY BLUE ARTILLERY',NUMA,
                & ' F-A-F GUIDED AMMO TYPE',ITYPE
              CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
              CALL OUTPUT2 (ITIME,1,IB,0)
217 CONTINUE
              IBSYS(IB,2)=IBSYS(IB,2)-NKILLS
              IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
              ELSE
                PROB=RND0
                IF (PROB.LE.PKIT.AND.IBSYS(IB,2).GT.0) THEN
                  C TARGET KILLED
                  & WRITE (10,') 'BLUE',IB,' KILLED BY BLUE ARTILLERY',NUMA,
                    & ' F-A-F GUIDED AMMO TYPE',ITYPE
                  IBSYS(IB,2)=IBSYS(IB,2)-1
                  IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
                  CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
                  CALL OUTPUT2 (ITIME,1,IB,0)
                  ENDF
                  NSHOTS=NSHOTS-1
                ENDF
221 CONTINUE
                NCYC=NCYC+1
                IF (NSHOTS.GT.0) GO TO 215
300 CONTINUE
                ENDF

          IF (IATYPE.EQ.2) THEN
            C EVALUATE UNGUIDED AREA MUNITIONS SUCH AS EXPLOSIVE ROUNDS (he)
            C AND BOMBLET DISPENSING CARGO PROJECTILES (dpcrm)
            C AGAIN, THESE MUNITIONS HAVE NO FRIENDS
            C FIND ALL RED AND BLUE TARGETS IN THE BASKET AREA PLUS CEP ERROR
            C AND EVALUATE FRACTIONAL COVERAGE.
            NTR=0
            C FIND ALL RED AND BLUE TARGETS IN BASKET AREA
            DO 310 IT=1,NR
              IF (IRSYS(IT,2).LE.0) GO TO 310
              XR=RXA(IT,1)*1000.
              YR=RYA(IT,2)*1000.
              R=((XR-XT)**2+(YR-YT)**2)**.5
              IF (R.LE.RADIUS) THEN
                C THIS RED VEHICLE CAN BE TARGETED
                ITARGR(IT)=1
                NTR=NTR+1
                ENDF
310 CONTINUE
                NTB=0
                DO 311 IT=1,NB
                  IF (IBSYS(IT,2).LE.0) GO TO 311
                  XR=BXA(IT,1)*1000.

```

```

              YR=BYA(IT,2)*1000.
              R=((XR-XT)**2+(YR-YT)**2)**.5
              IF (R.LE.RADIUS) THEN
                C THIS BLUE VEHICLE CAN BE TARGETED
                ITARGB(IT)=1
                NTB=NTB+1
                ENDF
311 CONTINUE
                NTOTAL=NTR+NTB
                IF (NTOTAL.LE.0) GO TO 400

            C FIND SHOT DENSITY IN BEATEN AREA
            NSHOTS=NUMR*NTUBES*NSUBS
            IF (NSHOTS.EQ.0) GO TO 400
            SDENS=REAL(NSHOTS)/BEATAREA

            C WRITE (',') 'NTR,NTB,NSHOTS,SDENS'
            C WRITE (',') 'NTR,NTB,NSHOTS,SDENS'

            DO 320 IR=1,NR
              IF (ITARGR(IR).NE.1) GO TO 320
              C SHOOT IT
              JSYS=IBSYS(IR,1)
              PKRAD=ARTPKB(JSYS,KASYS,ITYPE)
              PKAREA=PI*PKRAD*PKRAD
              EXVAL=PKAREA*SDENS
              PKIT=1-EXP(-EXVAL)

            C WRITE (',') 'PKRAD,PKAREA,EXVAL,PKIT'
            C WRITE (',') 'PKRAD,PKAREA,EXVAL,PKIT'

            IF (JSYS.GT.10) THEN
              C COUNTER-BATTERY SITUATION
              JTUBES=IBSYS(IR,2)
              JASYS=JSYS-10
              DTUBES=REAL(IABTUBE(JASYS))
              NKILLS=0
              CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
                &MILTYPE,IATYPE)
              DO 315 INK=1,NKILLS
                WRITE (10,') 'RED',IR,' KILLED BY BLUE ARTILLERY',NUMA,
                  & ' SUBMUNITION AMMO TYPE',ITYPE
                CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
                CALL OUTPUT2 (ITIME,2,IR,0)
315 CONTINUE
                IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
                IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
                ELSE
                  PROB=RND0
                  IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
                    C TARGET KILLED
                    & WRITE (10,') 'RED',IR,' KILLED BY BLUE ARTILLERY',NUMA,
                      & ' SUBMUNITION AMMO TYPE',ITYPE
                    IRSYS(IR,2)=IRSYS(IR,2)-1
                    IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
                    CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
                    CALL OUTPUT2 (ITIME,2,IR,0)
                    ENDF
320 CONTINUE
                    DO 321 IB=1,NB
                      IF (ITARGB(IB).NE.1) GO TO 321
                      C SHOOT IT
                      JSYS=IBSYS(IB,1)
                      PKRAD=ARTPKB(JSYS,KASYS,ITYPE)
                      PKAREA=PI*PKRAD*PKRAD
                      EXVAL=PKAREA*SDENS
                      PKIT=1-EXP(-EXVAL)
                      IF (JSYS.GT.10) THEN
                        C COUNTER-BATTERY SITUATION
                        JTUBES=IBSYS(IB,2)
                        JASYS=JSYS-10
                        DTUBES=REAL(IABTUBE(JASY9))
                        NKILLS=0
                        CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
                          &MILTYPE,IATYPE)
                        DO 316 INK=1,NKILLS
                          WRITE (10,') 'BLUE',IB,' KILLED BY BLUE ARTILLERY',NUMA,
                            & ' SUBMUNITION AMMO TYPE',ITYPE
                          CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
                          CALL OUTPUT2 (ITIME,1,IB,0)
316 CONTINUE
                          IBSYS(IB,2)=IBSYS(IB,2)-NKILLS
                          IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
                          ELSE
                            PROB=RND0
                            IF (PROB.LE.PKIT.AND.IBSYS(IB,2).GT.0) THEN
                              C TARGET KILLED
                              & WRITE (10,') 'BLUE',IB,' KILLED BY BLUE ARTILLERY',NUMA,
                                & ' SUBMUNITION AMMO TYPE',ITYPE
                              IBSYS(IB,2)=IBSYS(IB,2)-1
                              IF (IBSYS(IB,2).LT.0) IBSYS(IB,2)=0
                              CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
                              CALL OUTPUT2 (ITIME,1,IB,0)
                              ENDF
321 CONTINUE

```

```

400  CONTINUE
      ENDOF
      MFUB(N,1)=0
      MFUB(N,2)=0
      MFUB(N,3)=0
      MFUB(N,4)=0
      MFUB(N,5)=0
      XYFUB(N,1)=0
      XYFUB(N,2)=0
C    DRAW FIRE MISSION IMPACT ZONE ON ACTIVE GRID
      CALL DRAWART (1,XT,YT,RADIUS,IATYPE)
      ENDOF
500  CONTINUE
C    NOW DO RED FIRE MISSIONS
      DO 1500 N=1,NFMS/2
      IF (MFUR(N,1).EQ.0) GO TO 1500
      IETA=MFUR(N,5)
      IF (IETA.GT.0) THEN
C    NOT ARRIVED ON TARGET YET
      MFUR(N,5)=MFUR(N,5)-IDTIME
      GO TO 1500
      ELSE
C    THIS MISSION HAS COME DUE
      NUMA=MFUR(N,1)
      XA=RXYA(NUMA,1)*1000
      YA=RXYA(NUMA,2)*1000
      KASYS=IRSYS(NUMA,1)-10
      NTUBES=IRSYS(NUMA,2)
      NUMR=MFUR(N,2)
      ITYPE=MFUR(N,3)
      MILTYPE=0
      IF (IAFAFR(KASYS,ITYPE).EQ.1) MILTYPE=1
      IATYPE=IATYPR(KASYS,ITYPE)
      IF (MILTYPE.EQ.1) IATYPE=0
      MAXRNG=IARNGR(KASYS,ITYPE,1)
      MAXCEP=IARNGR(KASYS,ITYPE,3)
      NSUBS=IARNGR(KASYS,ITYPE,4)
      IF (NSUBS.LE.0) NSUBS=1
      PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
      BEATAREA=REAL(MFUR(N,4))
      XT=XYFUR(N,1)*1000
      YT=XYFUR(N,2)*1000
C    ADJUST IMPACT POINT BASED ON CEP AND RANDOM NUMBER
      RANGE=(XT-XA)**2+(YT-YA)**2**.5
      CEPM=REAL(MAXCEP)*RANGE/REAL(MAXRNG)
      RDEP=.573*CEPM
      SIGRD=RDEP/.6745
      CALL CEP (SIGMA)
      SIGRD=SIGRD*SIGMA
      XT=XT+SIGRD
      YT=YT+SIGRD
      ACEP=P1*SIGRD*SIGRD
      IF (BEATAREA.LT.ACEP) BEATAREA=ACEP
      IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
      RADIUS=(BEATAREA/P1)**.5
C    WRITE (',' ) ACEP,PATAREA,BEATAREA,RADIUS
C    WRITE (',' ) ACEP,PATAREA,BEATAREA,RADIUS
      DO 1103 IK=1,10
1103  IGUIDE(IK)=MILAR(N,IK)
      IF (IATYPE.EQ.3) THEN
C    CREATE ARTILLERY SMOKE CLOUD
C
      RADIUS=ARTPKR(1,KASYS,ITYPE)
      A1=P1*RADIUS*RADIUS
      AT=A1*NUMR*NTUBES
      IF (BEATAREA.GE.AT) THEN
      RADIUS=(AT/P1)**.5
      ELSE
      RADIUS=(BEATAREA/P1)**.5
      ENDOF
      TSMK=ARTPKR(2,KASYS,ITYPE)
      CALL ARTSMOKE (ITIME,XT,YT,RADIUS,TSMK)
      ENDOF
      IF (MILTYPE.EQ.1) THEN
C    EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C    TARGET LIST
C    LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C    IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C    BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.
C    FIND ALL BLUE TARGETS IN BASKET AREA
      DO 1110 IT=1,NB
      IF (IBSYS(IT,2).LE.0) GO TO 1110
      XR=RXYA(IT,1)*1000
      YR=RXYA(IT,2)*1000
      R=((XR-XT)**2+(YR-YT)**2)**.5
      IF (R.LE.RADIUS) THEN
C    THIS RED VEHICLE CAN BE TARGETED
      ITARGB(IT)=1
      ENDOF
1110  CONTINUE
C    COMPARE TARGET LIST AGAINST AVAILABLE TARGETS
      NSHOTS=NUMR*NTUBES*NSUBS
1115  DO 1120 IT=1,10
C    IF NO SHOTS LEFT, SKIP IT
      IF (NSHOTS.LE.0) GO TO 1120
      NT=IGUIDE(IT)
C    IF NO TARGET ON LIST, SKIP IT
      IF (NT.EQ.0) GO TO 1120
C    IF ALREADY KILLED SKIP IT
      IF (IBSYS(NT,2).LE.0) THEN
      IGUIDE(IT)=0
      GO TO 1120
      ENDOF
C    SEE IF THIS TARGET IS IN BASKET AND THEN SHOOT IT
      IF (ITARGB(NT).EQ.0) THEN
      IGUIDE(IT)=0
      GO TO 1120
      ENDOF
C    SHOOT IT
      JSYS=IBSYS(NT,1)
      PKIT=ARTPKR(JSYS,KASYS,ITYPE)
      IF (JSYS.GT.10) THEN
C    COUNTER-BATTERY SITUATION
      JTUBES=IBSYS(NT,2)
      JASYS=JSYS-10
      DTUBES=REAL(IABTUBE(JASYS))
      NKILLS=0
      CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOTS,PKIT,
      &MILTYPE,IATYPE)
      DO 1116 INK=1,NKILLS
      WRITE (10,') 'BLUE 'NT' KILLED BY RED ARTILLERY 'NUMA,
      &' M-I-L AMMO TYPE 'ITYPE
      CALL WARTY (ITIME,1,NT,2,NUMA,ITYPE)
      CALL OUTPUT2 (ITIME,1,NT,0)
1116  CONTINUE
      IBSYS(NT,2)=IBSYS(NT,2)-NKILLS
      IGUIDE(IT)=0
      ITARGB(NT)=0
      ELSE
      PROB=RND0
      IF (PROB.LE.PKIT) THEN
C    TARGET KILLED
      WRITE (10,') 'BLUE 'NT' KILLED BY RED ARTILLERY 'NUMA,
      &' M-I-L AMMO TYPE 'ITYPE
      IBSYS(NT,2)=IBSYS(NT,2)-1
      IF (IBSYS(NT,2).LT.0) IBSYS(NT,2)=0
C    REMOVE FROM TARGET LIST, SINCE KILLED
      IGUIDE(IT)=0
      ITARGB(NT)=0
      CALL WARTY (ITIME,1,NT,2,NUMA,ITYPE)
      CALL OUTPUT2 (ITIME,1,NT,0)
      ENDOF
      NSHOTS=NSHOTS-1
      ENDOF
1120  CONTINUE
      NUM=0
C    IF ANY SHOTS REMAIN, GO DOWN TARGETABLE LIST AND SHOOT AT THEM
      IF (NSHOTS.GE.1) THEN
      DO 1130 IT=1,NB
      IF (NUM.EQ.10) GO TO 1130
      IF (NUM.EQ.NSHOTS) GO TO 1130
      IF (ITARGB(IT).GT.0) THEN
      NUM=NUM+1
      IGUIDE(NUM)=IT
      ENDOF
1130  CONTINUE
      ENDOF
      IF (NUM.GT.0) GO TO 1115
      ENDOF
      IF (IATYPE.EQ.1) THEN
C    EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C    THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C    PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C    ROUNDS
C    FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C    PROBABILITIES OF BEING TARGETED AND KILLED
C    REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISN'T, FRIENDLY
C    VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C    IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C    CHECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C    THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C    FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.
C
      NTR=0
C    FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
      DO 1210 IT=1,NR
      XR=RXYA(IT,1)*1000
      YR=RXYA(IT,2)*1000
      R=((XR-XT)**2+(YR-YT)**2)**.5
      IF (R.LE.RADIUS) THEN
C    THIS RED VEHICLE CAN BE TARGETED
      ITARGB(IT)=1
      NTR=NTR+1

```

```

1210 CONTINUE
      NTB=0
      DO 1211 IT=1,NB
        XR=BXIA(IT,1)*1000
        YR=BYIA(IT,2)*1000
        R=((XR-XT)**2+(YR-YT)**2)**.5
        IF (R.LE.RADIUS) THEN
C      THIS BLUE VEHICLE CAN BE TARGETED
          ITARGB(IT)=1
          NTB=NTB+1
        ENDF
      CONTINUE
      NTOTAL=NTR+NTB
      IF (NTOTAL.EQ.0) GO TO 1300

C      FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
      NSHOTS=NUMR*NTUBES*NSUBS
      SEACH=REAL(NSHOTS)/REAL(NTOTAL)

C      BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C      THAT A VEHICLE WILL BE WITHIN RANGE OF AT LEAST ONE SUBMUNITION
      P1=1-EXP(-SEACH)

C      LIMIT SHOOT OUT TO NSUBS CYCLES, SINCE THE UNIFORMITY OF
C      THE PATTERN AND TARGET DENSITIES ARE UNKNOWN.
C      IT IS REASONABLE TO ASSUME THAT ANY ONE TARGET WILL BE
C      WITHIN THE ATTACK RANGE OF ANY ONE MISSILE.

      NCYC=0
1215 IF (NCYC.GT.NSUBS) GO TO 1300

      DO 1220 IR=1,NR
        IF (NSHOTS.LE.0) GO TO 1300
        IF (ITARGB(IR).NE.1) GO TO 1220
        PROB=RND0
        IF (PROB.GT.P1) GO TO 1220
C      SHOOT IT
        JSYS=IRSYS(IR,1)
        PKIT=ARTPKR(JSYS,KASYS,ITYPE)
        IF (JSYS.GT.10) THEN
C      COUNTER-BATTERY SITUATION
          JTUBES=IRSYS(IR,2)
          JASYS=JSYS-10
          DTUBES=REAL(IARTUBE(JASYS))
          NKILLS=0
          NSHOT=NSUBS

C      WRITE (',') 'CALL COUNTERB, NKILLS, JTUBES, DTUBES, RADIUS,
C      & NSHOT, PKIT, MILTYPE, IATYPE'
C      WRITE (',') NKILLS, JTUBES, DTUBES, RADIUS,
C      & NSHOT, PKIT, MILTYPE, IATYPE

          CALL COUNTERB (NKILLS, JTUBES, DTUBES, RADIUS, NSHOT, PKIT,
& MILTYPE, IATYPE)
C      WRITE (',') NKILLS'
C      WRITE (',') NKILLS

          NSHOTS=NSHOTS-NSUBS-NSHOT)
          DO 1216 INK=1,NKILLS
            WRITE (10,') 'RED', IR, 'KILLED BY RED ARTILLERY', NUMA,
              ' F-A-F GUIDED AMMO TYPE', ITYPE
            CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
            CALL OUTPUT2 (ITIME,2,IR,0)
1216 CONTINUE
            IRSYS(IR,2)=IRSYS(IR,2)-NKILLS
            IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
            ELSE
              PROB=RND0
              IF (PROB.LE.PKIT.AND.IRSYS(IR,2).GT.0) THEN
C      TARGET KILLED
                WRITE (10,') 'RED', IR, 'KILLED BY RED ARTILLERY', NUMA,
                  ' F-A-F GUIDED AMMO TYPE', ITYPE
                IRSYS(IR,2)=IRSYS(IR,2)-1
                IF (IRSYS(IR,2).LT.0) IRSYS(IR,2)=0
                CALL WARTY (ITIME,2,IR,1,NUMA,ITYPE)
                CALL OUTPUT2 (ITIME,2,IR,0)
              ENDF
              NSHOTS=NSHOTS-1
            ENDF
1220 CONTINUE

            DO 1221 IB=1,NB
              IF (NSHOTS.LE.0) GO TO 1300
              IF (ITARGB(IB).NE.1) GO TO 1221
              PROB=RND0
              IF (PROB.GT.P1) GO TO 1221
C      SHOOT IT
              JSYS=IRSYS(IB,1)
              PKIT=ARTPKR(JSYS,KASYS,ITYPE)
              IF (JSYS.GT.10) THEN
C      COUNTER-BATTERY SITUATION
                JTUBES=IRSYS(IB,2)
                JASYS=JSYS-10
                DTUBES=REAL(IARTUBE(JASYS))
                NKILLS=0
                NSHOT=NSUBS
                CALL COUNTERB (NKILLS, JTUBES, DTUBES, RADIUS, NSHOT, PKIT,
& MILTYPE, IATYPE)
                NSHOTS=NSHOTS-NSUBS-NSHOT)
              DO 1217 INK=1,NKILLS

```

```

              WRITE (10,') 'BLUE', IB, 'KILLED BY RED ARTILLERY', NUMA,
                ' F-A-F GUIDED AMMO TYPE', ITYPE
              CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
              CALL OUTPUT2 (ITIME,1,IB,0)
1217 CONTINUE
              IRSYS(IB,2)=IRSYS(IB,2)-NKILLS
              IF (IRSYS(IB,2).LT.0) IRSYS(IB,2)=0
              ELSE
                PROB=RND0
                IF (PROB.LE.PKIT.AND.IRSYS(IB,2).GT.0) THEN
C      TARGET KILLED
                  WRITE (10,') 'BLUE', IB, 'KILLED BY RED ARTILLERY', NUMA,
                    ' F-A-F GUIDED AMMO TYPE', ITYPE
                  IRSYS(IB,2)=IRSYS(IB,2)-1
                  IF (IRSYS(IB,2).LT.0) IRSYS(IB,2)=0
                  CALL WARTY (ITIME,1,IB,1,NUMA,ITYPE)
                  CALL OUTPUT2 (ITIME,1,IB,0)
                ENDF
                NSHOTS=NSHOTS-1
              ENDF
1221 CONTINUE

              NCYC=NCYC+1
              IF (NSHOTS.GT.0) GO TO 1215

1300 CONTINUE

            ENDF

            IF (IATYPE.EQ.2) THEN
C      EVALUATE UNGUIDED AREA MUNITIONS SUCH AS EXPLOSIVE ROUNDS (he)
C      AND BOMBLET DISPENSING CARGO PROJECTILES (dplcm)
C      AGAIN, THESE MUNITIONS HAVE NO FRIENDS
C      FIND ALL RED AND BLUE TARGETS IN THE BASKET AREA PLUS CEP ERROR
C      AND EVALUATE FRACTIONAL COVERAGE.
              NTR=0
C      FIND ALL RED AND BLUE TARGETS IN BASKET AREA
              DO 1310 IT=1,NR
                IF (IRSYS(IT,2).LE.0) GO TO 1310
                XR=RXIA(IT,1)*1000
                YR=RYIA(IT,2)*1000
                R=((XR-XT)**2+(YR-YT)**2)**.5
                IF (R.LE.RADIUS) THEN
C      THIS RED VEHICLE CAN BE TARGETED
                  ITARGR(IT)=1
                  NTR=NTR+1
                ENDF
1310 CONTINUE

                NTB=0
                DO 1311 IT=1,NB
                  IF (IRSYS(IT,2).LE.0) GO TO 1311
                  XR=RXIA(IT,1)*1000
                  YR=RYIA(IT,2)*1000
                  R=((XR-XT)**2+(YR-YT)**2)**.5

C      WRITE (',') 'BLUE, X Y R RADIUS'
C      WRITE (',') IT,XR,YR,R,RADIUS

                IF (R.LE.RADIUS) THEN
C      WRITE (',') 'YES'

C      THIS BLUE VEHICLE CAN BE TARGETED
                  ITARGB(IT)=1
                  NTB=NTB+1
                ENDF
1311 CONTINUE
                NTOTAL=NTR+NTB

C      WRITE (',') 'NTOTAL,NR,NB'
C      WRITE (',') 'NTOTAL,NTR,NTB'

                IF (NTOTAL.LE.0) GO TO 1400

C      FIND SHOT DENSITY IN BEATEN AREA
                NSHOTS=NUMR*NTUBES*NSUBS
                IF (NSHOTS.EQ.0) GO TO 1400

                SDENS=REAL(NSHOTS)/BEATEAREA

C      WRITE (',') 'NTR,NTB,NSHOTS,SDENS'
C      WRITE (',') NTR,NTB,NSHOTS,SDENS

              DO 1320 IR=1,NR
                IF (ITARGR(IR).NE.1) GO TO 1320
C      SHOOT IT
                JSYS=IRSYS(IR,1)
                PKRAD=ARTPKR(JSYS,KASYS,ITYPE)
                PKAREA=PI*PKRAD*PKRAD
                EXVAL=PKAREA*SDENS
                PKIT=1-EXP(-EXVAL)

C      WRITE (',') 'PKRAD,PKAREA,EXVAL,PKIT'
C      WRITE (',') PKRAD,PKAREA,EXVAL,PKIT

                IF (JSYS.GT.10) THEN
C      COUNTER-BATTERY SITUATION
                  JTUBES=IRSYS(IR,2)

```

```

JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NKILLS=0
CALL COUNTER8 (NKILLS, JTUBES, DTUBES, RADIUS, NSHOTS, PKIT,
&MILTYPE, IATYPE)
DO 1315 INK=1, NKILLS
WRITE (10, *) 'RED', IR, ' KILLED BY RED ARTILLERY ', NUMA,
&
SUBMUNITION AMMO TYPE ', ITYPE
CALL WARTY (ITIME, 2, IR, 1, NUMA, ITYPE)
CALL OUTPUT2 (ITIME, 2, IR, 0)
1315 CONTINUE
IRSYS(IR, 2)=IRSYS(IR, 2)-NKILLS
IF (IRSYS(IR, 2), LT, 0) IRSYS(IR, 2)=0
ELSE
PROB=RND0
IF (PROB, LE, PKIT, AND, IRSYS(IR, 2), GT, 0) THEN
C TARGET KILLED
WRITE (10, *) 'RED', IR, ' KILLED BY RED ARTILLERY ', NUMA,
&
SUBMUNITION AMMO TYPE ', ITYPE
IRSYS(IR, 2)=IRSYS(IR, 2)-1
IF (IRSYS(IR, 2), LT, 0) IRSYS(IR, 2)=0
CALL WARTY (ITIME, 2, IR, 1, NUMA, ITYPE)
CALL OUTPUT2 (ITIME, 2, IR, 0)
ENDIF
ENDIF
1320 CONTINUE
DO 1321 IB=1, NB
IF (ITARGB(IB), NE, 1) GO TO 1321
C SHOOT IT
JSYS=IBSYS(IB, 1)
PKRAD=ARTPKR(JSYS, KASYS, ITYPE)
PKAREA=PI*PKRAD*PKRAD
EXVAL=PKAREA*SOENS
PKIT=1-EXP(-EXVAL)
C WRITE (10, *) 'PKRAD, PKAREA, EXVAL, PKIT'
C WRITE (10, *) 'PKRAD, PKAREA, EXVAL, PKIT'
IF (JSYS, GT, 10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IBSYS(IB, 2)
JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NKILLS=0
CALL COUNTER8 (NKILLS, JTUBES, DTUBES, RADIUS, NSHOTS, PKIT,
&MILTYPE, IATYPE)
DO 1316 INK=1, NKILLS
WRITE (10, *) 'BLUE', IB, ' KILLED BY RED ARTILLERY ', NUMA,
&
SUBMUNITION AMMO TYPE ', ITYPE
CALL WARTY (ITIME, 1, IB, 1, NUMA, ITYPE)
CALL OUTPUT2 (ITIME, 1, IB, 0)
1316 CONTINUE
IBSYS(IB, 2)=IBSYS(IB, 2)-NKILLS
IF (IBSYS(IB, 2), LT, 0) IBSYS(IB, 2)=0
ELSE
PROB=RND0
IF (PROB, LE, PKIT, AND, IBSYS(IB, 2), GT, 0) THEN
C TARGET KILLED
WRITE (10, *) 'BLUE', IB, ' KILLED BY RED ARTILLERY ', NUMA,
&
SUBMUNITION AMMO TYPE ', ITYPE
IBSYS(IB, 2)=IBSYS(IB, 2)-1
IF (IBSYS(IB, 2), LT, 0) IBSYS(IB, 2)=0
CALL WARTY (ITIME, 1, IB, 1, NUMA, ITYPE)
CALL OUTPUT2 (ITIME, 1, IB, 0)
ENDIF
ENDIF
1321 CONTINUE
1400 CONTINUE
ENDIF
MFUR(N, 1)=0
MFUR(N, 2)=0
MFUR(N, 3)=0
MFUR(N, 4)=0
MFUR(N, 5)=0
XYFUR(N, 1)=0
XYFUR(N, 2)=0
C DRAW FIRE MISSION IMPACT ZONE ON ACTIVE GRID
CALL DRAWART (2, XT, YT, RADIUS, IATYPE)
ENDIF
1500 CONTINUE
RETURN
END
SUBROUTINE CEP (SIGMA)
C THIS SUBROUTINE DETERMINES FOR THIS ONE EVENT THE RANGE AND
C DEFLECTION IMPACT POINT ERROR FOR THE ARTILLERY BARRAGE BASED
C ON THE CEP INPUT.
C THIS IMPACT ERROR IS DETERMINED BY TAKING SUCCESSIVE RANDOM
C NUMBERS (FROM A FLAT PROBABILITY CURVE), (UP TO 7 CYCLES).
C THE RANDOM NUMBER IS EVALUATED FOR BEING ABOVE OR BELOW 0.5
C (THE 50% CHANCE POINT), AND THEN ASSIGNED TO A SIGMA BASKET.

```

C ALL SIGMA BASKETS ARE OF EQUAL AREA (THEREFORE, EQUAL PROBABILITY).  
C SINCE THE PROBABILITY DENSITY IS A BELL CURVE, THE SIGMA SPACING  
C IS UNEVEN, ALTHOUGH THE PROBABILITIES ARE THE SAME. THIS RESULTS  
C IN THE IMPACT ERROR BEING ASSIGNED TO A SIGMA BASKET WHICH  
C FOLLOWS THE NORMAL PROBABILITY CURVE.  
C SIGMA BASKETS ARE DEFINED TO WITHIN 3% ERROR AT SMALL SIGMAS TO  
C .8% ERROR AT LARGE SIGMAS (MORE BASKETS ARE NEEDED AS THE CURVE  
C THINS OUT AT THE TAILS, SO THAT ULTIMATE IMPACT POINT ERROR ERRORS  
C ARE REDUCED.

DIMENSION SB(20), RL(20)

CHARACTER\*7 RLL, LTEST

SB(1)=0.04  
SB(2)=0.12  
SB(3)=0.198  
SB(4)=0.2775  
SB(5)=0.3625  
SB(6)=0.4475  
SB(7)=0.535  
SB(8)=0.6275  
SB(9)=0.725  
SB(10)=0.8275  
SB(11)=0.98  
SB(12)=1.125  
SB(13)=1.235  
SB(14)=1.425  
SB(15)=1.605  
SB(16)=1.77  
SB(17)=1.925  
SB(18)=2.07  
SB(19)=2.285  
SB(20)=2.71

RL(1)='1000000'  
RL(2)='1000100'  
RL(3)='1001000'  
RL(4)='1001100'  
RL(5)='1010000'  
RL(6)='1010100'  
RL(7)='1011000'  
RL(8)='1011100'  
RL(9)='1100000'  
RL(10)='1100100'  
RL(11)='1101000'  
RL(12)='1101100'  
RL(13)='1110000'  
RL(14)='1110100'  
RL(15)='1111000'  
RL(16)='1111010'  
RL(17)='1111100'  
RL(18)='1111101'  
RL(19)='1111110'  
RL(20)='1111111'

DO 100 I=1, 7

P=RND0  
IF (P, LE, 0.50) THEN  
C ERROR IS NEGATIVE, ASSIGN VALUE OF 0  
L(I)=0  
ELSE  
C ERROR IS POSITIVE, ASSIGN VALUE OF 1  
L(I)=1  
ENDIF

100 CONTINUE

NFLAG=1

IF (L(1:1), EQ, 0) THEN  
NFLAG=-1  
L(1:1)='1'  
ENDIF

SIGMA=0

DO 200 I=1, 20  
LTEST=RL(I)  
IF (L, LE, 14) THEN  
IF (L(1:5), EQ, LTEST(1:5)) SIGMA=SB(I)  
ENDIF  
IF (L, EQ, 15, OR, L, EQ, 16) THEN  
IF (L(1:6), EQ, LTEST(1:6)) SIGMA=SB(I)  
ENDIF  
IF (L, GE, 17) THEN  
IF (L(1:7), EQ, LTEST(1:7)) SIGMA=SB(I)  
ENDIF

200 CONTINUE

SIGMA=SIGMA\*NFLAG

RETURN  
END

SUBROUTINE ARTSMOKE (ITIME, XT, YT, RADIUS, TSMK)

C CREATES ARTILLERY SMOKE PUFFS



```

COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX

```

```

C ARTILLERY SMOKE CLOUDS
  IF (NART.GT.0) THEN
C TEST TO SEE IF SMOKE ARRAY NEEDS TO BE COMPRESSED BACK
C TO COLUMN 1
  ITEST=1-IART2
  IF (ITEST.GT.NARTMX) THEN
C COMPRESS ARRAY BEFORE STARTING
    J=0
    DO 110 I=IART1,IART2
      J=J+1
      PART(J,1)=PART(I,1)
      PART(J,2)=PART(I,2)
      TAART(J,1)=TAART(I,1)
      TAART(J,2)=TAART(I,2)
110 CONTINUE
    IART1=1
    IART2=J
    ENDIF
  ENDIF

```

```

C CREATE NEW ARTILLERY SMOKE PUFF AFTER IART2
  IART2=IART2+1
  K=IART2
  PART(K,1)=XT/1000.
  PART(K,2)=YT/1000.
  TAART(K,1)=TSMK
  TAART(K,2)=RADIUS
  NART=IART2-IART1+1
C
  RETURN
  END

```

```

C SUBROUTINE PARTSMK
  BASED ON SUBROUTINE PENG
  C PLOTS ARTILLERY SMOKE PUFFS
  COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,
& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX
  COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
  COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
  COMMON/EARPUFFS/IAERASE,PAERASE(99,3)
C
  ISKIP=0
C
  DRAWS A GREEN CIRCLE FOR EACH PUFF LOCATION.
  AND SAVES THIS PUFF FOR LATER ERASING.
  RADIUS IS BASED ON PUFF TIME WHEN AT MAX ZOOM.
  AT MIN ZOOM, CIRCLE IS DRAWN LARGER SO YOU CAN SEE IT.
  CALL FACTOR (ZFACT)
  GXUNIT=11.0/NX*SCFACT
  GYUNIT=8.5/NY
  XUNIT=GXUNIT/SCALE
  YUNIT=GYUNIT/SCALE
  IAERASE=0
  DO 100 I=IART1,IART2
    RADIUS=TAART(I,2)*XUNIT*1.0/1000.
    X=PART(I,1)*XUNIT
    Y=PART(I,2)*YUNIT
    IF (XLEN.GT.0) THEN
      CALL FACTOR (ZFACTN)
      RADIUS=RADIUS*ZFACT/ZFACTN
    ENDIF
    CALL NEWPEN (10)
    IF ((X-15.*RADIUS).LE.0.0) ISKIP=1
    IF ((Y-15.*RADIUS).LE.0.0) ISKIP=1
    IF (XLEN.GT.0) THEN
      CALL ZOOMIT (X,Y,ISKIP)
      IF ((Y-RADIUS).GT.(95*YMAX)) ISKIP=1
      IF ((X-2.*RADIUS).LE.0.0) ISKIP=1
      IF ((Y-2.*RADIUS).LE.0.0) ISKIP=1
    ENDIF
    IF (ISKIP.EQ.1) GO TO 100
C DRAW CIRCLE
    CALL CIRCLE (X,Y,RADIUS)
    IAERASE=IAERASE+1
    PAERASE(IAERASE,1)=PART(I,1)
    PAERASE(IAERASE,2)=PART(I,2)
    PAERASE(IAERASE,3)=TAART(I,2)
100 CONTINUE
  RETURN
  END

```

```

C SUBROUTINE ASMASK (XS,YS,RMAX1,RMAX2,RMAX3)
  FIND ANGLE AND RANGE TO ALL ARTILLERY SMOKE PUFFS WITHIN
  MAX RANGE OF THIS SYSTEM SO THAT MASKING CAN BE EVALUATED
  COMMON/PUFFS/IWSPEED,IWDIR,IDUST,EXPUFF,DTPUFF,TENDPUFF,
& IDUST1,IDUST2,NDUST,IENG1,IENG2,NENG,ISMK1,ISMK2,NSMK,

```

```

& IART1,IART2,NART,PTDUST(2400,3),PTENG(2400,3),PSMK(400,2),
& TASMK(400,2),PART(99,2),TAART(99,2),NDUSTMX,NENGMX,NSMKMX,
& NARTMX

```

```

COMMON/MASKAR/NARMSK,ARMSK(99,3)

```

```

C XS,YS - SHOOTER COORDINATES
  RMAX1,2,3 - MAX RANGE OF SYSTEM WEAPONS
  ARMSK(1,1) - X COORD OF EACH ARTY SMOKE PUFF
  ARMSK(1,2) - Y COORD OF EACH ARTY SMOKE PUFF
  ARMSK(1,3) - RADIUS FOR EACH ARTY SMOKE PUFF
  NARMSK -- NUMBER OF ARTILLERY SMOKE PUFFS WITHIN MAX RANGE
C

```

```

  PL=3.14159
  NARMSK=0
  IF (NART.GT.0) THEN
C CYCLE THROUGH ALL SMOKE PUFFS TO FILL MASKING ARRAY
  RMAXX=0
  IF (RMAX1.GT.RMAXX) RMAXX=RMAX1
  IF (RMAX2.GT.RMAXX) RMAXX=RMAX2
  IF (RMAX3.GT.RMAXX) RMAXX=RMAX3
  RMAXX=RMAXX/1000.
  DO 20 I=IART1,IART2
    XD=PART(I,1)
    YD=PART(I,2)
    RADIUS=PART(I,3)/1000.
    RANGE=((XD-XS)**2+(YD-YS)**2)**.5
    IF ((RANGE-RADIUS).GT.RMAXX) GO TO 20
    NARMSK=NARMSK+1
    ARMSK(NARMSK,1)=XD
    ARMSK(NARMSK,2)=YD
    ARMSK(NARMSK,3)=RADIUS
20 CONTINUE
  ENDIF
  RETURN
  END

```

```

C SUBROUTINE ASSEE (XS,YS,XT,YT,AZMUTH,JSKIP)
  EVALUATES ARTILLERY SMOKE MASKING BETWEEN SHOOTER AND TARGET
C

```

```

  XS,YS SHOOTER REAL WORLD COORDINATES
  XT,YT TARGET REAL WORLD COORDINATES
  AZIN AZIMUTH TO TARGET
  JSKIP FLAG TO SKIP THIS TARGET IF MASKED (-0 TO SKIP)
C

```

```

COMMON/ELEVA/CONT(20),IELEV(10000)
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT

```

```

COMMON/MASKAR/NARMSK,ARMSK(99,3)

```

```

DOUBLE PRECISION ANGLE,ANGLEI

```

```

  PL=3.14159
  IF (NARMSK.GT.0) THEN
C FIND SHOOTER AND TARGET GRID COORDINATES
  TAKE NEAREST INTEGER, SINCE GRIDS ARE LOCATED
  CENTER TO CENTER ABOUT THE SCALE.
  RRANGE=((YT-YS)**2+(XT-XS)**2)**.5
  IXS=NINT(XS/SCALE)
  IYS=NINT(YS/SCALE)
  IXT=NINT(XT/SCALE)
  IYT=NINT(YT/SCALE)
  IF (IXS.LT.1) IXS=1
  IF (IXT.LT.1) IXT=1
  IF (IYS.LT.1) IYS=1
  IF (IYT.LT.1) IYT=1
  IF (IXS.GT.NX) IXS=NX
  IF (IXT.GT.NX) IXT=NX
  IF (IYS.GT.NY) IYS=NY
  IF (IYT.GT.NY) IYT=NY
C FIND SHOOTER AND TARGET ELEVATION
  IS=NY*(IXS-1)+IYS
  SELEV=CONT(IELEV(IS))
  IT=NY*(IXT-1)+IYT
  TELEV=CONT(IELEV(IT))
C ADD 2.0 METER TO ELEVATIONS OF TARGET AND SHOOTER TO ACCOUNT
  FOR VEHICLE HEIGHT
  SELEV=SELEV+.002
  TELEV=TELEV+.002
C FIND ELEVATION ANGLE TO THIS TARGET
  TOP=TELEV-SELEV
  BOT=RRANGE
  ANGLE=ATAN(TOP/BOT)
C DO 200 I=1,NARMSK
  IF (JSKIP.EQ.0) GO TO 300
C

```

```

C CYCLE THROUGH ALL ARTY SMOKE PUFFS
  TO EVALUATE MASKING, FIND THE LEFT AND RIGHT AZIMUTH
  BRACKETING EACH PUFF WITHIN RANGE. DROP IF OUT OF ANGLE RANGE.
  THEN FIND THE ELEVATION ANGLE TO THE TOP OF THE PUFF.
  DROP PUFF IF NOT HIGH ENOUGH
C EVALUATE RANGE TO SMOKE PUFF
  XP=ARMSK(I,1)
  YP=ARMSK(I,2)

```

```

RADIUS=ARMSK(L3)
RP=(YP-YS)**2+(XP-XS)**2**.5
IF ((RP-RADIUS).GT.RRANGE) GO TO 200
CC EVALUATE ANGLES TO SMOKE PUFF
TOP=YP-YS
BOT=XP-XS
IF (TOP.EQ.0.0.AND.BOT.EQ.0.0) JSKIP=0
IF (BOT.EQ.0.0) BOT=BOT+.001
DIR=ATAN(ABS(TOP)/ABS(BOT))*180/P1
IF (BOT.LT.0.0) THEN
    IF (TOP.LT.0.0) DIR=DIR+180.
    IF (TOP.GT.0.0) DIR=180.-DIR
ELSE
    IF (TOP.LT.0.0) DIR=360.-DIR
ENDIF
C CONVERT THIS COORDINATE SYSTEM DIRECTION TO AN AZIMUTH
CALL FAZ (DIR,AZIN)
IF (RADIUS.LT.0.030) RADIUS=0.030
IF (RP.LT.RADIUS) RP=RADIUS
DELAZ=ASIN(RADIUS/RP)*180/P1
AZLFT=AZIN-DELAZ
AZRGT=AZIN+DELAZ
IF (AZLFT.LT.0.0) AZLFT=AZLFT+360.0
IF (AZRGT.LT.0.0) AZRGT=AZRGT+360.0
IF (AZLFT.GT.360.0) AZLFT=AZLFT-360.0
IF (AZRGT.GT.360.0) AZRGT=AZRGT-360.0
IF (AZLFT.GT.AZRGT) AZRGT=AZRGT+360.0
IF (AZIMUTH.LT.0.0) AZIMUTH=AZIMUTH+360.0
IF (AZIMUTH.GT.360.0) AZIMUTH=AZIMUTH-360.0
IF (AZIMUTH.LT.AZLFT.OR.AZIMUTH.GT.AZRGT) GO TO 200
CCCC EVALUATE HEIGHT OF SMOKE PUFF
C FIND ELEVATION OF THE GROUND UNDER THIS PUFF
IXP=NINT(XP/SCALE)
IYP=NINT(YP/SCALE)
IF (IXP.LT.1) IXP=1
IF (IYP.LT.1) IYP=1
IF (IXP.GT.NX) IXP=NX
IF (IYP.GT.NY) IYP=NY
IP=NY*(IXP-1)+IYP
PELEV=CONT(IELEV(IP))
C ADD TO THIS ELEVATION THE HEIGHT OF THE PUFF
C MAX PUFF HEIGHT IS 30 METERS (100 FEET)
HEIGHT=RADIUS
IF (RADIUS.GT.0.030) HEIGHT=0.030
PELEV=PELEV+HEIGHT
C FIND ELEVATION ANGLE TO THIS DUST PUFF
TOP=PELEV-SELEV
BOT=RP
ANGLE=ATAN(TOP/BOT)
IF (ANGLE.GT.ANGLE) JSKIP=0
200 CONTINUE
ENDIF
300 RETURN
END

SUBROUTINE DRAWART (KSIDE,XT,YT,RADIUS,IATYPE)
C BASED ON SUBROUTINE PENG
C PLOTS ARTILLERY FIRE MISSION ON ACTIVE GRID
COMMON/GRID/NX,NY,SCALE,SCFACT,PRFACT,ZFACT
COMMON/ZOOM/BLX,BLY,XLEN,ZFACTN,XMAX,YMAX,YUNIT
C IF (IATYPE.EQ.3) RETURN
ISKIP=0
CCCC DRAWS A COLORED CIRCLE (RED OR BLUE) FOR EACH MISSION LOCATION.
C CALL FACTOR (ZFACT)
GXUNIT=11.0/NX*SCFACT
GYUNIT=.85/NY
XUNIT=GXUNIT/SCALE
YUNIT=GYUNIT/SCALE
RADIUS=RADIUS*XUNIT*1.0/1000.
X=XT*XUNIT/1000.
Y=YT*YUNIT/1000.
IF (XLEN.GT.0.) THEN
    CALL FACTOR (ZFACTN)
    RADIUS=RADIUS*ZFACT/ZFACTN
ENDIF
IF (KSIDE.EQ.1) CALL NEWPEN (9)
IF (KSIDE.EQ.2) CALL NEWPEN (12)
IF ((X-15.*RADIUS).LE.0.0) ISKIP=1
IF ((Y-15.*RADIUS).LE.0.0) ISKIP=1
IF (XLEN.GT.0.) THEN
    CALL ZOOMIT (X,Y,ISKIP)
    IF ((Y-RADIUS).GT.(.95*YMAX)) ISKIP=1
    IF ((X-2.*RADIUS).LE.0.0) ISKIP=1
    IF ((Y-2.*RADIUS).LE.0.0) ISKIP=1
ENDIF
IF (ISKIP.EQ.1) GO TO 100
C DRAW CIRCLE
CALL CIRCLE (X,Y,RADIUS)
100 CONTINUE
RETURN

```

END

SUBROUTINE WARTY (KTIME,JSIDE,J,KSIDE,K,I,TYPE)

ARTILLERY KILLS DATA OUTPUT

```

CCCCCCCCC
KTIME      KILL TIME
JSIDE      TARGET SIDE (1 BLUE, 2 RED)
J          VEHICLE NUMBER
JSYS       SYSTEM NUMBER
KSIDE      SHOOTER SIDE
K          SHOOTER NUMBER
KSYS       SHOOTER ARTILLERY SYSTEM NUMBER
I,TYPE     ARTILLERY AMMUNITION NUMBER
CCCCCCCCC

```

COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),  
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR

CHARACTER\*30 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM  
CHARACTER\*16 BHIER,RHIER,IER  
CHARACTER\*24 BUNIT,RUNIT

```

IF (JSIDE.EQ.1) THEN
    JSYS=IBSYS(J,1)
ELSE
    JSYS=IRSYS(J,1)
ENDIF
IF (KSIDE.EQ.1) THEN
    KSYS=IBSYS(K,1)
ELSE
    KSYS=IRSYS(K,1)
ENDIF

```

1 FORMAT (8(14,3X))

WRITE (16,1) KTIME,JSIDE,J,JSYS,KSIDE,K,KSYS,I,TYPE

RETURN  
END

SUBROUTINE COUNTER8 (NKILLS,JTUBES,DTUBES,FADIUS,NSHOTS,PKIT,  
& MILTYPE,IATYPE)

C EVALUATES ARTILLERY FIRE ON ARTILLERY UNITS WHERE EACH  
C UNIT NUMBER REPRESENTS MULTIPLE TUBES OR VEHICLES

PI=3.14159

NTARG=1  
NKILLS=0

IF (JTUBES.EQ.0) RETURN

IF (JTUBES.EQ.2) THEN  
 IF (DTUBES.LE.RADIUS) NTARG=2  
ENDIF

IF (JTUBES.GE.3) THEN

```

C DEFINE A POLYGON WHERE EACH TUBE IS LOCATED THE SAME
C DISTANCE DTUBES FROM EACH OTHER TUBE
C FIND THE OUTSIDE RADIUS OF THIS POLYGON
A=DTUBES
ALPHA=2.*PI/REAL(JTUBES)
R=A/2/SIN(ALPHA/2.)

```

C IF THE MUNITION PATTERN IS AIMED AT ONE OF THE VERTICES,  
C THEN TO GET ALL OF THE TUBES IN THE PATTERN, THE PATTERN  
C RADIUS MUST BE TWICE THE POLYGON RADIUS.

C IF THE PATTERN IS AIMED AT THE CENTER OF THE POLYGON, THEN  
C ALL TUBES CAN BE HIT IF THE PATTERN RADIUS IS EQUAL TO THE  
C POLYGON RADIUS

C SINCE, WE DO NOT KNOW, ASSUME IT IS SOMEWHERE IN BETWEEN.  
C THE NUMBER OF TUBES WHICH CAN BE TARGETED EQUALS THE RATIO  
C OF THE TWO DISTANCES.

RATIO=RADIUS/R

```

IF (RATIO.GE.2.) THEN
    NTARG=JTUBES
ELSE
    NTARG=NINT(RATIO*REAL(JTUBES-1))
ENDIF

```

IF (NTARG.LT.1) NTARG=1  
IF (NTARG.GT.JTUBES) NTARG=JTUBES

ENDIF

```

C IF (MILTYPE.EQ.1.OR.IATYPE.EQ.1) THEN
C M-I-L OR F-A-F GUIDED MUNITION OR
C KSHOT=NSHOTS
C DO 10 I=1,KSHOT
C IF (NTARG.LT.1) RETURN
C PROB=RND()
C IF (PKIT.GE.PROB) THEN

```

```

      NKILLS=NKILLS+1
      NTARG=NTARG+1
    ENDOF
    NSHOTS=NSHOTS+1
10  CONTINUE
    ENDOF

    IF (ATYPE.EQ.2) THEN
C    UNGUIDED MUNITION OR SUBMUNITIONS
C    APPLY FRACTIONAL COVERAGE TO NUMBER OF TARGETS IN PATTERN AREA
      NKILLS=NINT(PKIT*REAL(NTARG))
    ENDOF

C    WRITE (":") NKILLS
C    WRITE (":") NKILLS

      RETURN
    END

SUBROUTINE FCEST (KSIDE,NUMA,KASYS,ITYPE,NUMR,JTARG,IAREA)
C
C    ESTIMATES FRACTIONAL COVERAGE OF AREA TARGETS WHEN SELECTING
C    AN ARTILLERY MISSION WHICH USES HE OR DPMN MUNITIONS
C
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
& .NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
& .IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
& .ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
& .ISCOOTB(10,4),ISCOOTR(10,4)

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*9 BNAME,RNAME,BANAME,RANAME
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

      PI=3.14159

      IF (KSIDE.EQ.1) THEN
C    BLUE
        NTUBES=IBSYS(NUMA,2)
        NSUBS=IARNGB(KASYS,ITYPE,4)
        MCEP=IARNGB(KASYS,ITYPE,3)
        MRNG=IARNGB(KASYS,ITYPE,1)
        IF (NSUBS.LE.0) NSUBS=1
        PATAREA=REAL(IARNGB(KASYS,ITYPE,5))
        XA=BXYA(NUMA,1)*1000.
        YA=BXYA(NUMA,1)*1000.
        XT=RXYA(JTARG,1)*1000.
        YT=RXYA(JTARG,2)*1000.
        JSYS=IRSYS(JTARG,1)
        PKRAD=ARTPKB(JSYS,KASYS,ITYPE)
      ELSE
C    RED
        NTUBES=IRSYS(NUMA,2)
        NSUBS=IARNGR(KASYS,ITYPE,4)
        MCEP=IARNGR(KASYS,ITYPE,3)
        MRNG=IARNGR(KASYS,ITYPE,1)
        IF (NSUBS.LE.0) NSUBS=1
        PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
        XA=RXYA(NUMA,1)*1000.
        YA=RXYA(NUMA,1)*1000.
        XT=BXYA(JTARG,1)*1000.
        YT=BXYA(JTARG,2)*1000.
        JSYS=IBSYS(JTARG,1)
        PKRAD=ARTPKR(JSYS,KASYS,ITYPE)
      ENDOF

      RANGE=((XT-XA)**2+(YT-YA)**2)**.5
      CEP=RANGE/REAL(MRNG)*REAL(MCEP)
      BEATAREA=REAL(IAREA)

      IF (BEATAREA.LT.PATAREA) BEATAREA=PATAREA
      RADIUS=(BEATAREA/PI)**.5
      IF (RADIUS.LT.CEP) THEN
        RADIUS=CEP
        BEATAREA=PI*RADIUS*RADIUS
      ENDOF

      NTR=0
C    FIND ALL RED AND BLUE TARGETS IN BASKET AREA
      DO 310 IT=1,NR
        IF (IRSYS(IT,2).LE.0) GO TO 310
        NT=IRSYS(IT,2)
        IF (NT.GT.1) THEN
          JASYS=IRSYS(IT,1)-10
          DTUBES=REAL(IARTUBE(JASYS))
          NSHOTS=NUMR*NTUBES*NSUBS
          NKILLS=0
          CALL COUNTERB (NKILLS,NT,DTUBES,RADIUS,NSHOTS,1,0,0,2)
          NT=NKILLS
          ENDOF
          XR=RXYA(IT,1)*1000.

```

```

          YR=RXYA(IT,2)*1000.
          R=((XR-XT)**2+(YR-YT)**2)**.5
          IF (R.LE.RADIUS) THEN
C    THIS RED VEHICLE CAN BE TARGETED
            NTR=NTR+1*NT
          ENDOF
310  CONTINUE

        NTB=0
        DO 311 IT=1,NB
          IF (IBSYS(IT,2).LE.0) GO TO 311
          NT=IBSYS(IT,2)

C    WRITE (":") 'BLUE'
C    WRITE (":") IT

          IF (NT.GT.1) THEN
            JASYS=IBSYS(IT,1)-10
            DTUBES=REAL(IABTUBE(JASYS))
            NSHOTS=NUMR*NTUBES*NSUBS

C    WRITE (":") 'NSHOTS, DTUBES, JASYS'
C    WRITE (":") 'NSHOTS,DTUBES,JASYS'

          NKILLS=0
          CALL COUNTERB (NKILLS,NT,DTUBES,RADIUS,NSHOTS,1,0,0,2)
          NT=NKILLS

C    WRITE (":") 'NKILLS'
C    WRITE (":") NT

          ENDOF
          XR=BXYA(IT,1)*1000.
          YR=BXYA(IT,2)*1000.
          R=((XR-XT)**2+(YR-YT)**2)**.5

C    WRITE (":") 'X Y R RADIUS'
C    WRITE (":") XR,YR,R,RADIUS

          IF (R.LE.RADIUS) THEN
C    WRITE (":") 'YES'

C    THIS BLUE VEHICLE CAN BE TARGETED
            NTB=NTB+1*NT
          ENDOF
311  CONTINUE

C    FIND SHOT DENSITY IN BEATEN AREA
        NSHOTS=NUMR*NTUBES*NSUBS

        IF (BEATAREA.LE.0.0) GO TO 400
        SDENS=REAL(NSHOTS)/BEATAREA

        PKAREA=PI*PKRAD*PKRAD
        EXVAL=PKAREA*SDENS
        PKIT=1-EXP(-EXVAL)
        FC=PKIT*100.

        WRITE (":") 'ESTIMATED FRACTIONAL COVERAGE'
        WRITE (":") 'BASED ON BEATEN AREA'
        IF (KSIDE.EQ.1) THEN
          WRITE (":") '# BLUE TARGETS, # RED TARGETS, F.C % ON RED'
        ELSE
          WRITE (":") '# BLUE TARGETS, # RED TARGETS, F.C % ON BLUE'
        ENDIF

        WRITE (":") NTB,NTR,FC
        WRITE (":")

400  CONTINUE
      RETURN
    END

SUBROUTINE PTEST (KSIDE,NUMA,KASYS,ITYPE,NUMR,JTARG,IAREA)
C
C    ESTIMATES POINT TARGET MUNITION EFFECTIVENESS WHEN SELECTING
C    AN ARTILLERY MISSION WHICH USES m-H OR f-H MUNITIONS
C
      COMMON/MOVE/BXYA(200,2),RXYA(200,2),IBSA(200,2),IRSA(200,2),
& L1BM,L2M,L3M,L1RM
      COMMON/FORCE/BHIER(200),RHIER(200),BUNIT(200),RUNIT(200),
& IRSYS(200,2),IRSYS(200,2),L1BF,L2F,L3F,L1RF,NB,NR
      COMMON/ARTYDAT/NBFA,NRFA,BANAME(10),RANAME(10),NAMB(10),NAMR(10)
& .NAMMOB(10,5),NAMMOR(10,5),IATYPB(10,5),IATYPR(10,5)
& .IAFAFB(10,5),IAFAFR(10,5),IARNGB(10,5,5),IARNGR(10,5,5),
& .ARTPKB(20,10,5),ARTPKR(20,10,5),IABTUBE(10),IARTUBE(10),
& .ISCOOTB(10,4),ISCOOTR(10,4)

      CHARACTER*90 L1BF,L2F,L3F,L1RF,L1BM,L2M,L3M,L1RM
      CHARACTER*9 BNAME,RNAME,BANAME,RANAME
      CHARACTER*16 BHIER,RHIER,IHIER
      CHARACTER*24 BUNIT,RUNIT

      PI=3.14159

      IF (KSIDE.EQ.1) THEN
C    BLUE

```

```

NTUBS=IBSYS(NUMA,2)
NSUBS=IARNGR(KASYS,ITYPE,4)
MCEP=IARNGR(KASYS,ITYPE,3)
MRNG=IARNGR(KASYS,ITYPE,1)
IF (NSUBS.LE.0) NSUBS=1
PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
XA=BXA(NUMA,1)*1000.
YA=BYA(NUMA,1)*1000.
XT=RXA(JTARG,1)*1000.
YT=RYA(JTARG,2)*1000.
JSYS=IRSYS(JTARG,1)
MLTYPE=IAFAFB(KASYS,ITYPE)
LATYPE=IATYPR(KASYS,ITYPE)
ELSE
C RED
NTUBS=IRSYS(NUMA,2)
NSUBS=IARNGR(KASYS,ITYPE,4)
MCEP=IARNGR(KASYS,ITYPE,3)
MRNG=IARNGR(KASYS,ITYPE,1)
IF (NSUBS.LE.0) NSUBS=1
PATAREA=REAL(IARNGR(KASYS,ITYPE,5))
XA=BXA(NUMA,1)*1000.
YA=BYA(NUMA,1)*1000.
XT=RXA(JTARG,1)*1000.
YT=RYA(JTARG,2)*1000.
JSYS=IBSYS(JTARG,1)
MLTYPE=IAFAFR(KASYS,ITYPE)
LATYPE=IATYPR(KASYS,ITYPE)
ENDIF

RANGE=((XT-XA)**2+(YT-YA)**2)**.5
CEP=RANGE/REAL(MRNG)*REAL(MCEP)
BEATAREA=REAL(PATAREA)

IF (BEATAREALLT.PATAREA) BEATAREA=PATAREA
RADIUS=(BEATAREA/P)**.5
IF (RADIUS.LT.CEP) THEN
RADIUS=CEP
BEATAREA=P*RADIUS*RADIUS
ENDIF

IF (KSIDE.EQ.1) THEN

C BLUE
IF (MLTYPE.EQ.1) THEN
LATYPE=0
C EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C TARGET LIST
C LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

NSHOTS=NUMR*NTUBS*NSUBS
NTR=0
C FIND ALL RED TARGETS IN BASKET AREA
DO 110 IT=1,NR
IF (IRSYS(IT,2).LE.0) GO TO 110
XR=RXA(IT,1)*1000.
YR=RYA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS RED VEHICLE CAN BE TARGETED
NA=1
JSYS=IRSYS(IT,1)
PKIT=ARTPKB(JSYS,KASYS,ITYPE)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IRSYS(IT,2)
JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NSHOT=NSHOTS
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
&MLTYPE,LATYPE)
NA=NKILLS
ENDIF
NTR=NTR+1*NA
ENDIF
110 CONTINUE
FC=PKIT*100.

WRITE (**) 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
WRITE (**) '# RED TGT, % KILLS'
WRITE (**) NTR,FC

ENDIF

IF (LATYPE.EQ.1) THEN
C EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C ROUNDS
C FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C PROBABILITIES OF BEING TARGETED AND KILLED
C REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISNT, FRIENDLY
C VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.

NTR=0
C FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
DO 210 IT=1,NR
XR=RXA(IT,1)*1000.
YR=RYA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS RED VEHICLE CAN BE TARGETED
NA=1
JSYS=IRSYS(IT,1)
PKIT=ARTPKB(JSYS,KASYS,ITYPE)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IRSYS(IT,2)
JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NKILLS=0
NSHOT=NSUBS
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
&MLTYPE,LATYPE)
NA=NKILLS
ENDIF
NTR=NTR+1*NA
ENDIF
210 CONTINUE

NTB=0
DO 211 IT=1,NB
XR=BXA(IT,1)*1000.
YR=BYA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS BLUE VEHICLE CAN BE TARGETED
NA=1
JSYS=IBSYS(IT,1)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IBSYS(IT,2)
JASYS=JSYS-10
DTUBES=REAL(IABTUBE(JASYS))
NKILLS=0
NSHOT=NSUBS
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
&MLTYPE,LATYPE)
NA=NKILLS
ENDIF
NTB=NTB+1*NA
ENDIF
211 CONTINUE
NTOTAL=NTR+NTB

IF (NTOTAL.LE.0) GO TO 300

C FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
NSHOTS=NUMR*NTUBS*NSUBS
SEACH=REAL(NSHOTS)/REAL(NTOTAL)
C BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C THAT A VEHICLE WILL BE WITHIN RANGE OF ONE SUBMUNITION
P1=1.-EXP(-SEACH)
P2=1.-{1.-PKIT}*NSUBS
300 CONTINUE
FC=P1*P2*100.

WRITE (**) 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
WRITE (**) '# BLUE TGT, # RED TGT, % KILLS RED'
WRITE (**) NTB,NTR,FC

ENDIF
ELSE
C RED
IF (MLTYPE.EQ.1) THEN
LATYPE=0
C EVALUATE MAN-IN-THE-LOOP TERMINAL GUIDANCE ON DESIGNATED
C TARGET LIST
C LISTED TARGETS MUST BE IN THE BASKET AREA PLUS CEP ERROR
C IF THESE TARGETS ARE NO LONGER THERE, ANY SHORTFALL WILL
C BE MADE UP BY FINDING ANY OTHER ENEMY TARGETS IN AREA.

NSHOTS=NUMR*NTUBS*NSUBS
NTB=0
C FIND ALL BLUE TARGETS IN BASKET AREA
DO 410 IT=1,NB
IF (IBSYS(IT,2).LE.0) GO TO 410
XB=BXA(IT,1)*1000.
YB=BYA(IT,2)*1000.
R=((XB-XT)**2+(YB-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS BLUE VEHICLE CAN BE TARGETED
NA=1
JSYS=IBSYS(IT,1)
PKIT=ARTPKB(JSYS,KASYS,ITYPE)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IBSYS(IT,2)
JASYS=JSYS-10

```

```

DTUBES=REAL(IABTUBE(JASYS))
NSHOT=NSHOTS
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
&MILTYPE,IATYPE)
NA=NKILLS
ENDIF
NTB=NTB+1*NA
ENDIF
410 CONTINUE
FC=PKIT*100.

WRITE (',') 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
WRITE (',') '# BLUE TGT, % KILLS'
WRITE (',') NTB,FC

ENDIF

IF (IATYPE.EQ.1) THEN
C EVALUATE POINT TARGET MUNITIONS/SUBMUNITIONS
C THESE ARE FIRE-AND-FORGET TERMINALLY GUIDED
C PROJECTILES/SUBMUNITIONS SUCH AS SADARM OR GUIDED MORTAR
C ROUNDS
C FIND ALL VEHICLES IN THE BASKET AREA AND EVALUATE THEIR
C PROBABILITIES OF BEING TARGETED AND KILLED
C REMEMBER ... UNLESS THERE IS IFF, WHICH THERE ISN'T, FRIENDLY
C VEHICLES MAY ALSO BE TARGETED, SO FIND ALL POTENTIAL VEHICLES
C IN BASKED AREA, INCLUDING KILLED VEHICLES, SINCE HOW THE
C HECK WILL A MUNITION LIKE THIS KNOW THE DIFFERENCE ?
C THE FINER POINTS OF THESE ISSUES WILL HAVE TO BE MODELED LATER.
C FOR NOW, THESE FAF GUIDED PROJECTILES ARE BASICALLY DUMB.
C
NTR=0
C FIND ALL RED AND BLUE TARGETS IN BASKET AREA, EVEN DEAD ONES
DO 510 IT=1,NR
XR=RXVA(IT,1)*1000.
YR=RYVA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS RED VEHICLE CAN BE TARGETED
NA=1
JSYS=RSYS(IT,1)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IRSYS(IT,2)
JASYS=JSYS-10
DTUBES=REAL(IARTUBE(JASYS))
NKILLS=0
NSHOT=NSUBS*NUMR*NTUBES
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
&MILTYPE,IATYPE)
NA=NKILLS
ENDIF
NTR=NTR+1*NA
ENDIF
510 CONTINUE

NTB=0
DO 511 IT=1,NB
XR=BXVA(IT,1)*1000.
YR=BYVA(IT,2)*1000.
R=((XR-XT)**2+(YR-YT)**2)**.5
IF (R.LE.RADIUS) THEN
C THIS BLUE VEHICLE CAN BE TARGETED
NA=1
JSYS=BSYS(IT,1)
PKIT=ARTPKR(JSYS,KASYS,IATYPE)
IF (JSYS.GT.10) THEN
C COUNTER-BATTERY SITUATION
JTUBES=IBSYS(IT,2)
JASYS=JSYS-10
DTUBES=REAL(IABTUBE(JASYS))
NKILLS=0
NSHOT=NSUBS*NUMR*NTUBES
CALL COUNTERB (NKILLS,JTUBES,DTUBES,RADIUS,NSHOT,1.0,
&MILTYPE,IATYPE)
NA=NKILLS
ENDIF
NTB=NTB+1*NA
ENDIF
511 CONTINUE
NTOTAL=NTR+NTB

IF (NTOTAL.LE.0) GO TO 600

C FIND EXPECTED NUMBER OF SHOTS PER AVAILABLE TARGET
NSHOTS=NUMR*NTUBES*NSUBS
SEACH=REAL(NSHOTS/REAL(NTOTAL))
C BASED ON POISSON'S DISTRIBUTION, FIND THE PROBABILITY
C THAT A VEHICLE WILL BE WITHIN RANGE OF AT LEAST ONE SUBMUNITION
P1=1.-EXP(-SEACH)
C PROBABILITY OF BEING KILLED IF IN RANGE OF ALL SUBMUNITIONS
C IN ONE MISSILE
P2=1.-((1.-PKIT)**NSUBS)
600 CONTINUE
FC=P1*P2*100.

WRITE (',') 'ESTIMATED NUMBER OF TARGETS IN PATTERN AREA'
WRITE (',') '# RED TGT, # BLUE TGT, % KILLS BLUE'
WRITE (',') NTR,NTB,FC

```